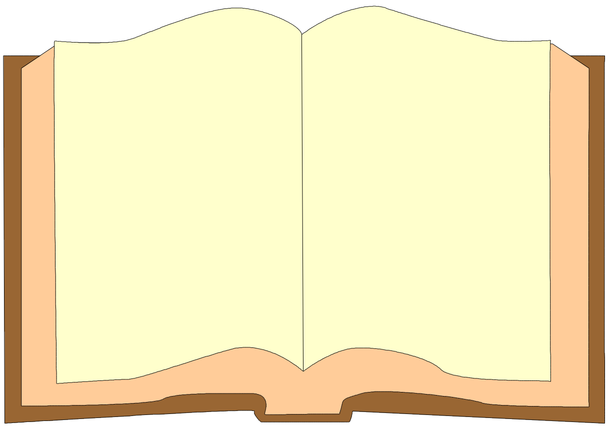
ＫＭＣ基本情報受験学習講座シリーズ

基礎理論

アルゴリズム



平成２８年１１月１日

加藤　正夫

アルゴリズム目次

探索のアルゴリズム

整列アルゴリズム

ファイル処理

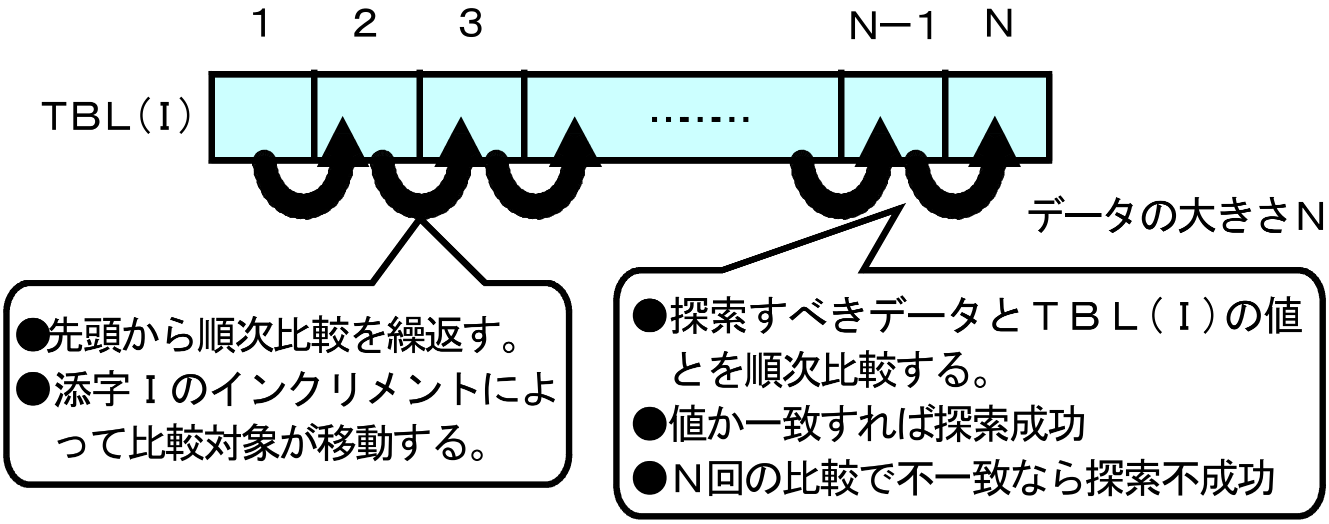
文字列処理

探索のアルゴリズム

◆線形探索法

◇線形探索法とは

　線形探索法は、配列の先頭から探索データを探し出すまで、または、配列の最後に至り探索データがないことが判るまで、配列の要素を１つずつ調べる方法である。



　探索は、目的の要素を発見するだけでなく、目的の要素の有無の判定も行う。

　対象の各要素の一部として含まれる探索キーと、目的の要素がもつキーとを比較することによって、探索を行う。

　線形探索には、先頭から行う探索と後方から行う探索がある。

　先頭から行う探索は配列の大きさＮを越えると終了し、後方からの探索は先頭の添字より小さくなると終了するる

◇線形探索法の基本手順（先頭からの探索）

①　初期条件の設定：１→Ｉ

後方からの探索の場合は、Ｎ→Ｉとなる。

②　添字の範囲のチェック：Ｉ＞Ｎ

後方からの探索の場合は、Ｉ＜１となる。

③　データの比較：ＴＢＬ（Ｉ）：Ｋ（探索データ）

④　添字のインクリメント：Ｉ＋１→Ｉ

後方からの探索の場合は、添字のディクリメントとなり、Ｉ－１→Ｉとなる。

⑤　終了条件の判定：ＴＢＬ（Ｉ）＝ＫまたはＩ＞Ｎ

後方からの探索の場合は、ＴＢＬ（Ｉ）＝ＫまたはＩ＜１

◇線形探索法の比較回数

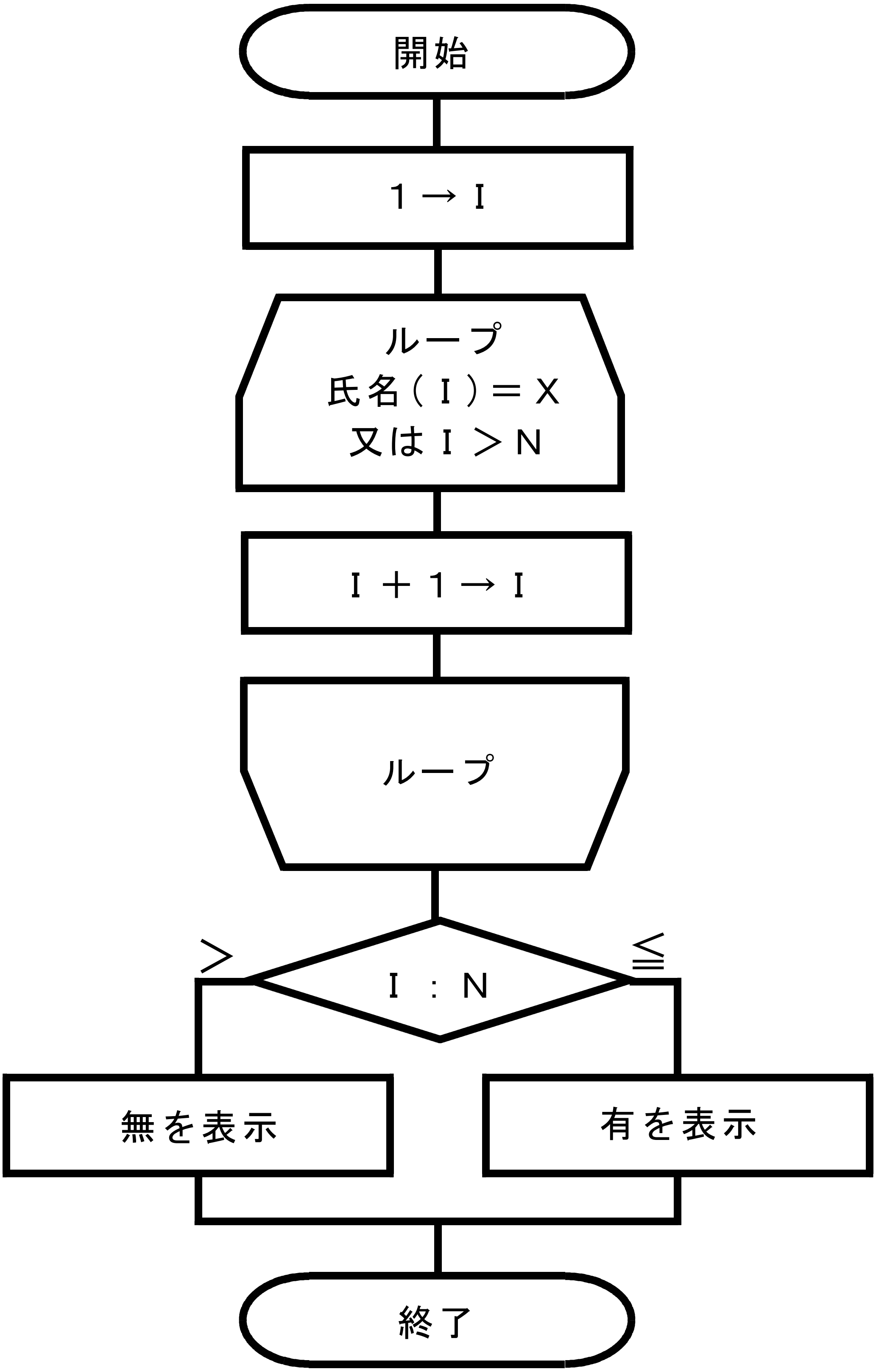
データの個数がＮの場合、比較回数は次のようになる。

平均比較回数は（Ｎ＋１）／２

最大比較回数はＮ回

Ｎが十分に大きい場合、平均比較回数はＮ／２となる。

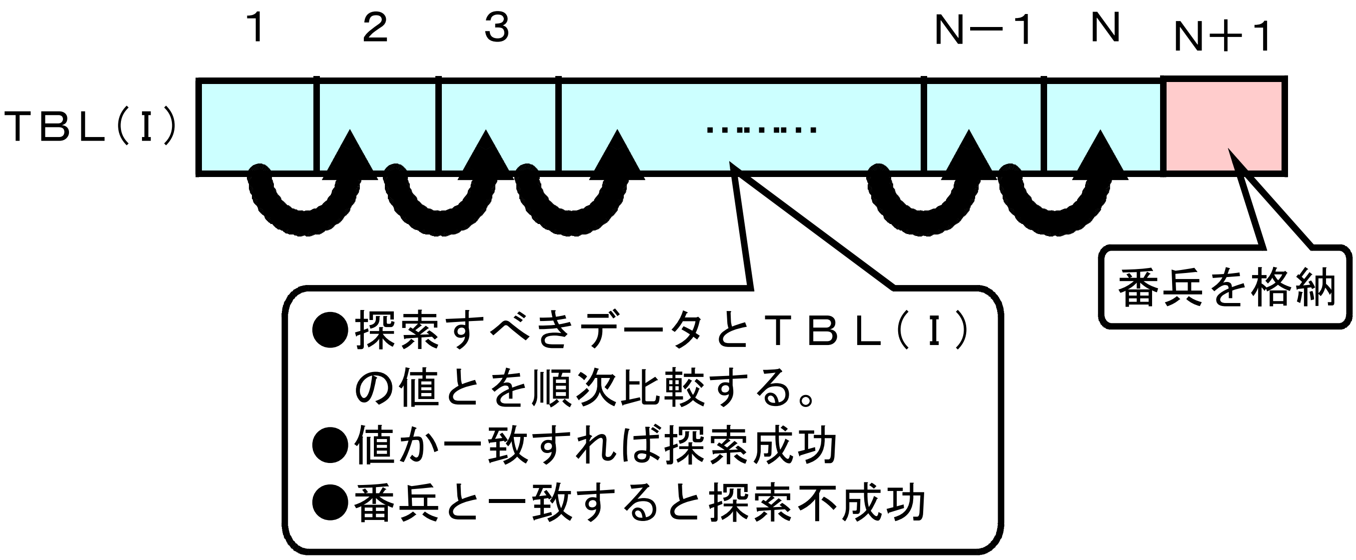
◇線形探索の流れ図



◇番兵を使用した線形探索法

　線形探索法には番兵という考え方がある。ループの終了条件を簡単にするために、探索するデータを番兵として最後のデータの次に置く。

　番兵を設定することによって、添字の範囲のチェックが不要となり、比較回数が減少する。



◇番兵を使用した線形探索の手順

①　初期条件の設定：１→Ｉ

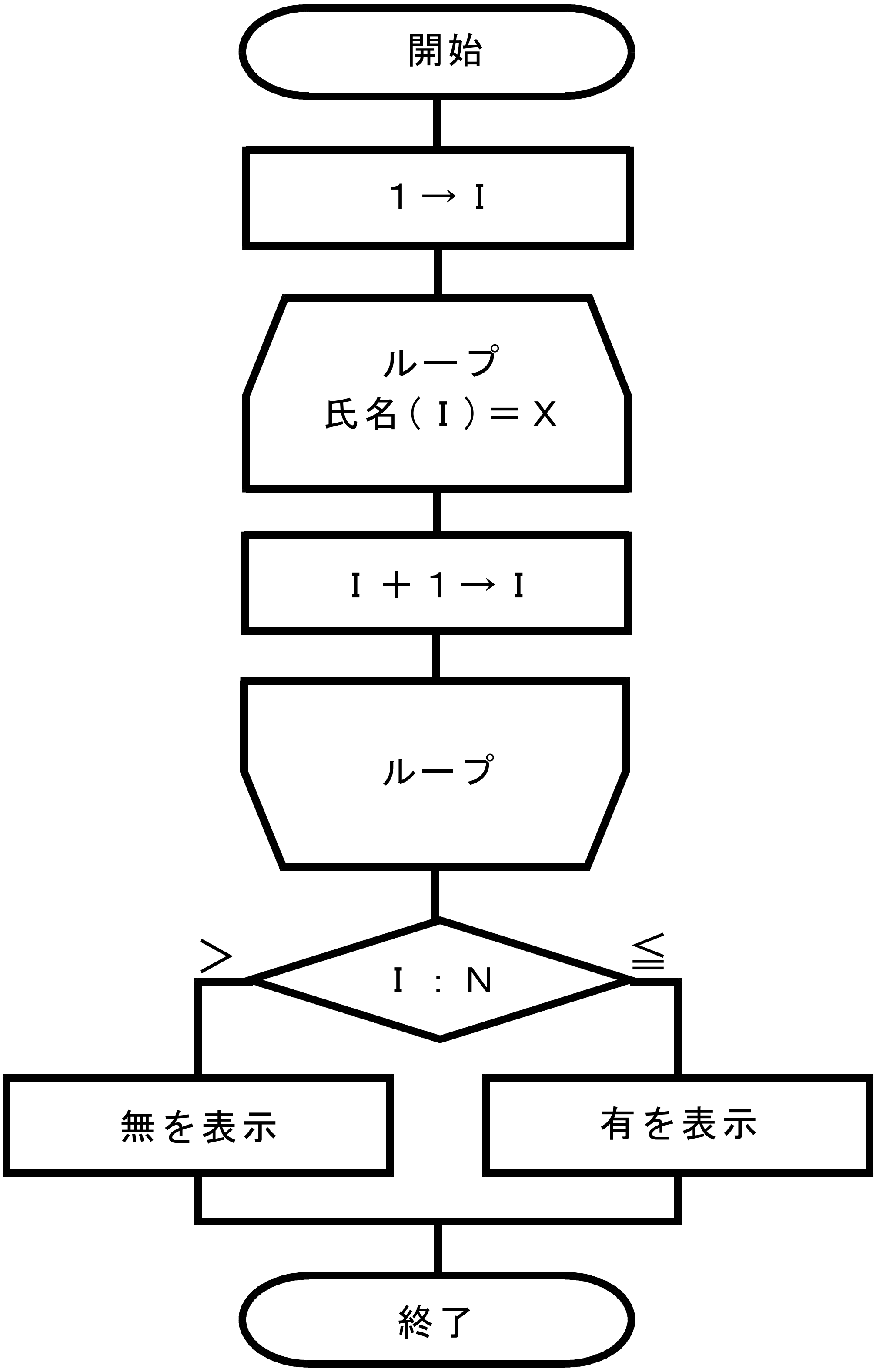
②　データの比較：ＴＢＬ(Ｉ)：Ｋ(探索データ)

③　添字のインクリメント：Ｉ＋１→Ｉ

④　終了条件の判定：ＴＢＬ(Ｉ)＝Ｋ

⑤　Ｉ＜Ｎの場合、探索成功。Ｉ＝Ｎの場合、探索不成功。

◇番兵を使用した線形探索の流れ図



◇線形探索法の通常と番兵の比較回数

配列に１０名の氏名が格納されている場合の通常の線形探索法と番兵を使用した線形探索法の最悪の場合の比較回数を求める。

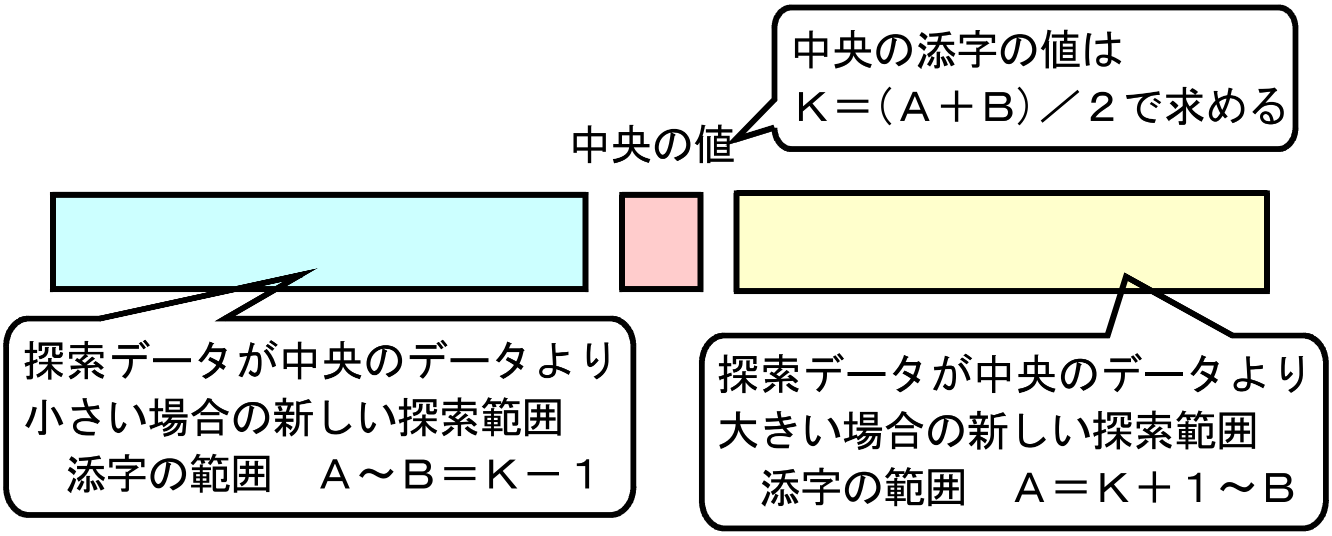
通常の比較回数３４に対して、番兵を使用した場合は、比較回数は２４かいとなる。

ステップの内容	通常の場合	番兵の場合
添字Ｉの初期設定	１	１
配列の領域Ｉ＞１０の判定	１１	０
氏名（Ｉ）と探索データＸの比較	１０	１１
Ｉ＋１→Ｉの演算	１０	１０
Ｉ＞１０の判定	１	１
結果の表示	１	１
全比較回数	３４	２４

番兵を使用した線形探索は、配列の領域の判定が不要になり、比較回数が減少する。

◆二分探索法

◇二分探索法とは



二分探索法は、比較対象のデータを配列の中央の添字の値に設定し、比較する。

一致しない場合、整列済み配列を２分割して、探索データがどちらの系列に属するかを調べ、中央の値と比較する。再び、２分割して属する範囲を調べる。この操作を繰り返してデータを探索する方法である。

探索の対象となる配列が、あらかじめ昇順（降順）に並んでいるときに用いることができる。

◇二分探索法の手順

①　対象の配列を昇順または降順に並べる。

②　探索する範囲、探索下限の添字Ａ、探索上限の添字Ｂを設定する。

③　比較対象の中央の要素の添字Ｋ＝[（Ａ＋Ｂ）／２]を決める。

④　Ｋの要素の値と探索する値が一致すると探索成功。探索を終了する。

⑤　Ｋの要素の値＞探索する値のときは、探索上限の添字Ｂ＝Ｋ－１を求め、Ａ～Ｂを新しい探索範囲とし、処理⑦に移る。

⑥　Ｋの要素の値＜探索する値のときは、探索下限の添字Ａ＝Ｋ＋１を求め、Ａ～Ｂを新しい探索範囲とし、処理⑦に移る。

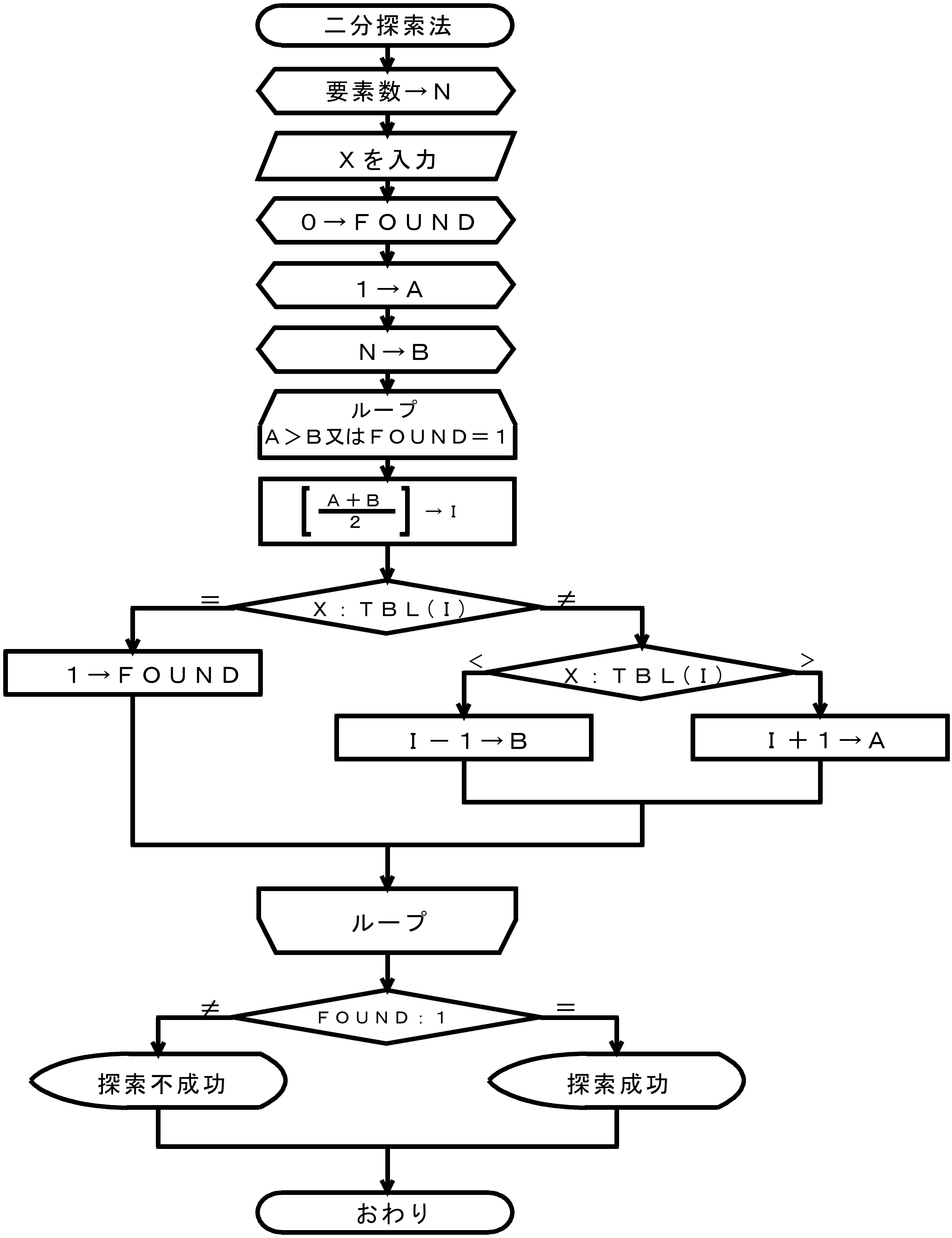
⑦　探索下限の添字Ａと探索上限の添字Ｂが、Ａ≦Ｂの間、③～⑥を繰り返す。

⑧　探索下限の添字Ａと探索上限の添字Ｂが、Ａ＞Ｂとなると探索不成功。対象の配列の中に探索する値が存在しないので探索を終了する。

◇二分探索法の終了条件

探索する範囲の探索下限の添字をＡ、探索上限の添字をＢとすると、Ａ＞Ｂになると、探索を終了する。Ａ＞Ｂであって、Ａ≧Ｂでないことに注意する必要がある。

◇二分探索法の流れ図


二分探索法
要素数→N
Xを入力
0→FOUND
1→A
N→B
ループ
A＞B又はFOUND＝1
A＋B
2
→I
＝
≠
X：TBL（I）
1→FOUND
＜
＞
X：TBL（I）
I－1→B
I＋1→A
ループ
≠
＝
FOUND：1
探索不成功
探索成功
おわり

◇二分探索法の比較回数

データ個数がＮの場合の比較回数は、次の通り。

①　平均比較回数は［$\log_2 N$］回となる。

②　最大比較回数［$\log_2 N$］＋１回となる。

平均比較回数は、探索するデータが見つかる場合の最大の比較回数であり、最大比較回数は、探索するデータが見つからない場合の比較回数である。

最大比較回数は、平均比較回数＋１となる。

◆探索アルゴリズムの計算量

◇アルゴリズムと計算量

アルゴリズムの構築では、次の事項を検討し、実行効率の向上を図る必要がある。

① 正しい理論に従って手順を組み立てる。

② わかりやすく修正しやすい手続きによって組み立てる。

③ 実行効率を最適にする。

④ アルゴリズムを改善する。

⑤ 多数の条件の整理や、条件判断の順序を最適にする。

同一のデータ構造を使用してもアルゴリズムが異なると実行効率が異なる。実行効率は命令実行数や比較回数で判断することができる。

◇線形探索法と二分探索法の最大比較回数

① 線形探索法の最大比較回数

データ数ｎの場合、線形探索の最大比較回数はｎ回である。

② 二分探索法の最大比較回数

データ数ｎの場合、二分探索法の平均比較回数をＫとすると、次の式が成り立つ

$$2^{K} \leqq n < 2^{K+1}$$

両辺の２を底とする対数をとると

$$\log_2 2^k \leqq \log_2 n < \log_2 2^{K+1}$$
$$K \leqq \log_2 n < K+1$$
$$K = [\log_2 n]$$

平均比較回数は$[\log_2 n]$となる。

平均比較回数よりも、更に、１回２分割の回数を増やすと、データ数がｎを超えるため、その時の比較回数が最大比較回数となり、

$$K+1 = [\log_2 n] + 1$$

となる。

$n = 10000 = 10^4$の時は

$$\begin{aligned}[\log_2 10^4] + 1 &= [4 / \log 2] + 1 \\ &= [4 / 0.301] + 1 \\ &= 13 + 1 \\ &= 14\end{aligned}$$

最大比較回数は１４回となる。

$n = 10^8$の時は

$$\begin{aligned}[\log_2 10^8] + 1 &= [8 / \log 2] + 1 \\ &= [8 / 0.301] + 1 \\ &= [26.6] + 1 \\ &= 26 + 1 = 27\end{aligned}$$

最大比較回数は２７回となる。

◇線形探索法、二分探索法、ハッシュ法の計算量

データ数に対する線形探索、二分探索、ハッシュ法の場合の最大比較回数を求めると、表のようになる。

	最大比較回数		
データ数	線形探索法	二分探索法	ハッシュ法
１０００	１０００	１０	1
１００００	１００００	１４	1
１０^8	１０^8	２７	1
N	N	$\log_2 N + 1$	1

データ数が大きくなると、二分探索法の最大比較数は、線形探索法の最大比較数に比べて大幅に少なくなる。

二分探索法の場合、データ数が１０００程度の場合、データ数が１０倍の１００００になると、線形探索法に比べて、最大比較回数は１．４倍の増加となる。

しかし、データ数が１万から１万倍の１億に増加した場合、二分探索法の比較回数は、線形探索法の最大探索回数に比べて、２倍程度に増加するだけである。

◆ハッシュ法

◇ハッシュ法の定義

配列、リスト、二分木のデータ構造は一定の順序で処理する必要があったが、ハッシュ法はデータの値から直接、格納位置を計算することができる。データの参照を高速に行う。

キー値Ｘを関数Ｈ（Ｘ）を用いて、配列の添字やアドレスに変換する。関数Ｈ（Ｘ）をハッシュ関数といい、ハッシュ関数が返す値をハッシュ値という。ハッシュ関数を用いて、データを格納する添字やアドレスを求めることをハッシングという。ハッシュ法の計算量は通常、Ｏ（１）である。

◇衝突（コリジョン）

異なったキー値から同じハッシュ値が求まることを衝突という。先に格納されていたデータをホームデータ、後に格納されるデータをシノニムデータという。

衝突が発生した場合、この衝突を回避する方法に、チェーン法とオープンアドレス法がある。

◇チェーン法

データを格納する場合、同じハッシュ値を持つデータをポインタを利用してリストにつなぐ方法である。

◇チェーン法のデータ格納手順

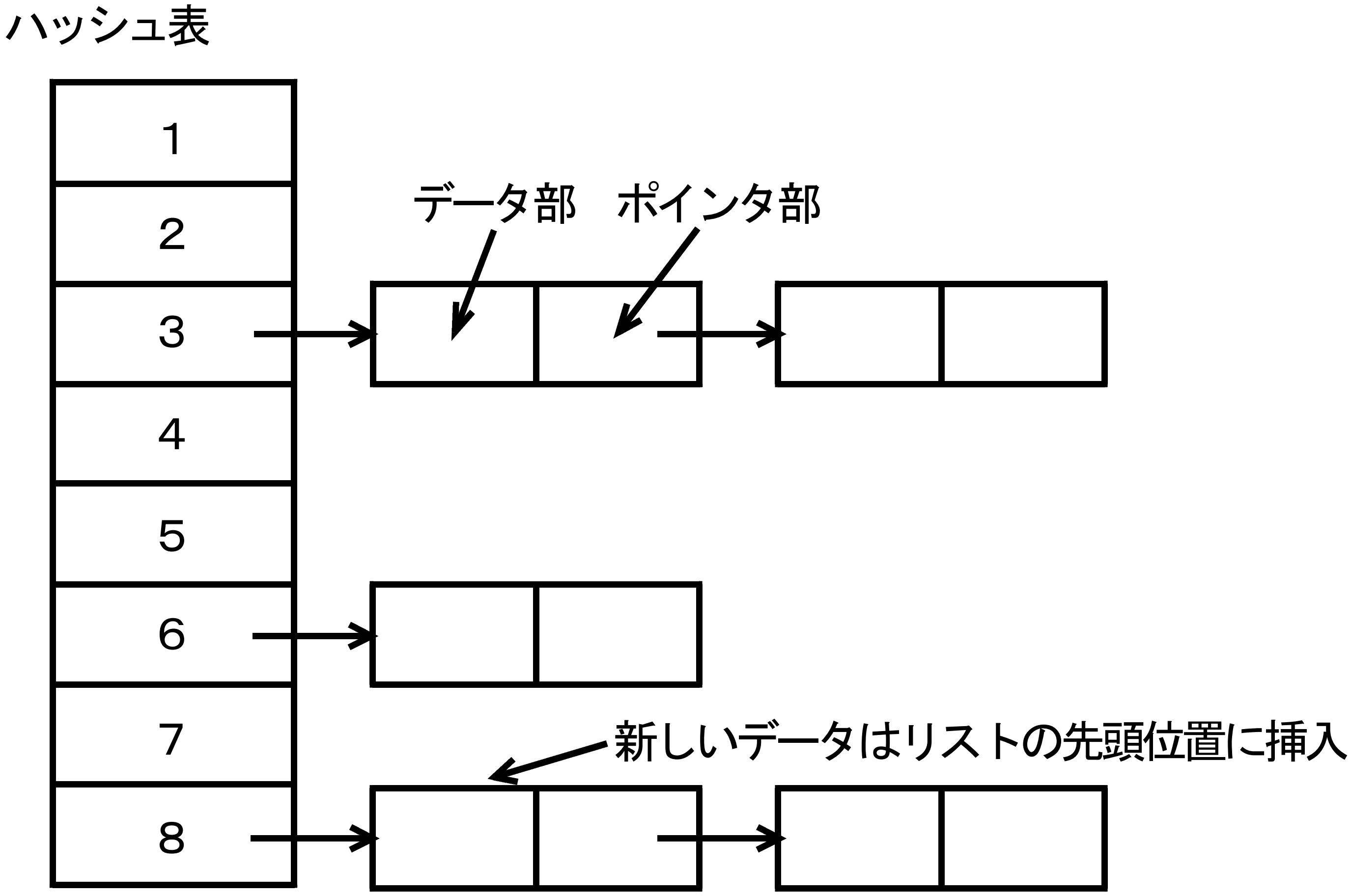
①　ハッシュ関数を用いてデータを格納する位置を計算する。

②　データの格納箇所が空であれば、データをそのアドレスに格

納し、ハッシュ表のポインタ部にそのアドレスを設定する。

③　空でなければ、リストの先頭に新しいデータを挿入する要領でデータを挿入する。

◇チェーン法のデータ探索手順



①　データとハッシュ関数からハッシュ値を求め、格納位置を知る。

②　ハッシュ表のアドレスから最初のデータを探し、探索するデータと比較する。

③　一致すれば探索成功。

④　不一致ならば、ポインタ部から次の比較データの格納位置を求め、探索データと比較する。データが見つかるか、データがなくなるまで同じ操作を繰り返す。

◇チェーン法探索の計算量

格納位置を決める計算量はＯ（１）、チェーンでつながっているデータを探索する線形探索はＯ（Ｎ／Ｂ）となり、探索全体ではＯ（１＋Ｎ／Ｂ）となる。Ｎはデータの数、Ｂは格納箇所の数である。

リストの先頭に格納する計算量はＯ（１）である。重複チェックを行う場合は全体でＯ（１＋Ｎ／Ｂ）となる。

◇オープンアドレス法

オープンアドレス法でデータを格納する場合、衝突が発生すると再ハッシュで対処し、データをハッシュ表の中に全て格納する方法である。

再ハッシュの方法は、「元のハッシュ値＋１」を用いる。

衝突すると、再ハッシュ値を求め、そのデータ部にデータが入っていると、「元のハッシュ値＋１」で次の格納位置を求める。空いている格納位置が見つかるまで再ハッシュを繰り返し位置が決まれば挿入する。

探索の場合も、ハッシュ値を求め格納位置を求めると、その位置のデータの値と比較して同じであれば、探索成功となる。

異なれば、「元のハッシュ値＋１」で再ハッシュして、次の格納位置のデータの値と比較する。データの値が一致するか、空のデータ部が現れると探索終了になる。

整列のアルゴリズム

◆整列のアルゴリズム

◇整列とは

整列は、データ列をある規則に従って並べ替えることである。小さい順に並び替えることを昇順に整列といい、大きい順に並び替えることを降順に整列という。

◇内部整列と外部整列

① 内部整列

内部整列は対象とするデータ群が主記憶装置にある場合の整列である。

② 外部整列

外部整列は対象とするデータ群が補助記憶装置にある場合の整列である。

◇整列の計算量

データを並べ替える場合、それぞれのアルゴリズムの考え方に基づいて、データの比較が行われる。この比較の回数を計算量という。

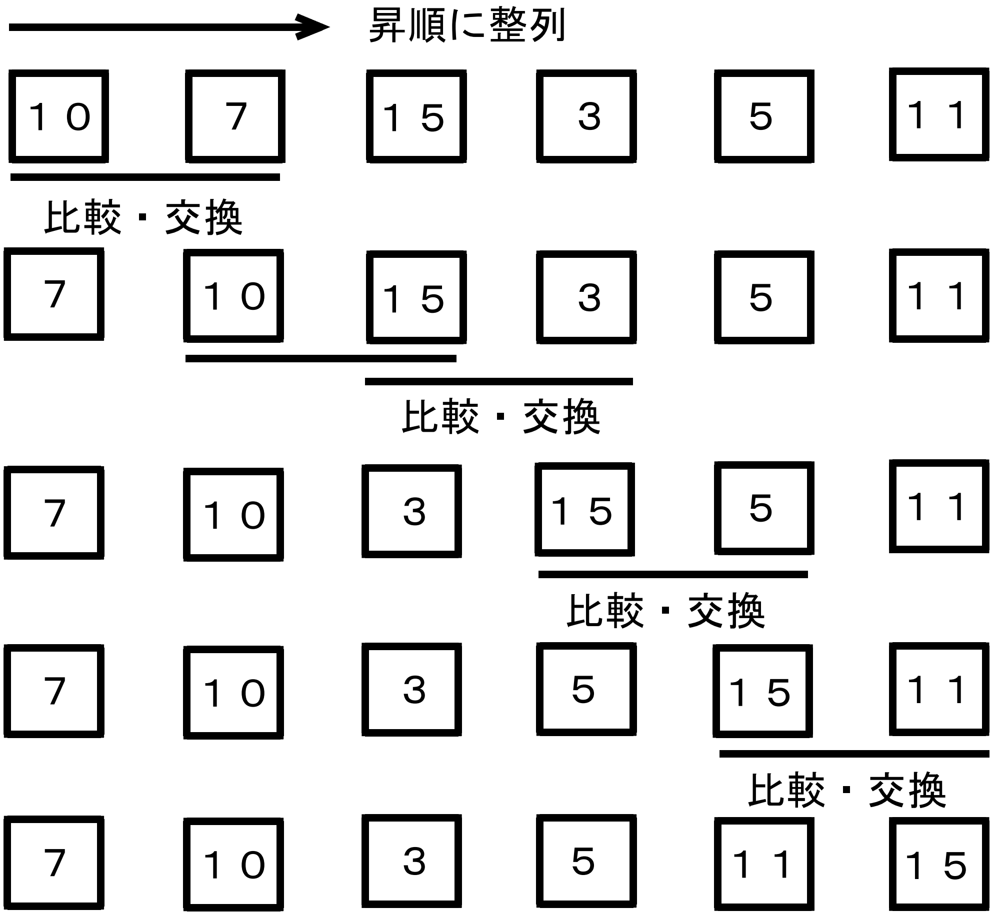
計算量の小さいアルゴリズムがＣＰＵを効率的に使用するアルゴリズムになる。アルゴリズムは常に計算量を問題にする。

◆基本交換法（バブルソート）

◇基本交換法

　基本交換法は配列の隣り合った２個のデータを比較して、大小の順序が違っていると並べ替えていく方法でデータを整列する。

◇基本交換法の手順



①　配列の先頭のデータと次のデータの大小を比較する。大小関係が異なると入れ替える。

②　２番目のデータと次のデータの大小を比較する。大小関係が異なると入れ替える。

③　順次、移動して、隣り合う２個のデータを比較しながら、大小関係が異なると入れ替える操作を繰り返す。

④　配列の最後まで来ると、１回目の比較を終了する。この１回目の比較で、最後の要素の値が決まる。

⑤　２回目も同様の比較を行い、２回目に比較した要素の中の最後尾の要素の値が決まる。

⑥　各回の操作で、その回の最後尾の要素の値が決まる。

⑦　すべての配列の要素が昇順または降順に並ぶと操作を完了する。

◇基本交換法の比較回数

対象のデータがｎ個ある場合、１回の操作でｎ－１回の比較を行い、最後尾の１個のデータの正しい位置が決まる。

次に、対象のデータｎ－１個に対して、ｎ－２回の操作を行い、最後尾の１個のデータの正しい位置が決まる。

この操作を順次繰り返して、ｎ－１回目に２個のデータを比較して最後尾のデータの正しい位置が決まる。

全部のデータの位置を決めるのに必要な比較回数は次の式で計算できる。

$$
\begin{aligned}
(n-1)+(n-2)+\cdots+2+1 &= n(n-1)/2 \\
&= (n^2-n)/2 \\
&\propto n^2
\end{aligned}
$$

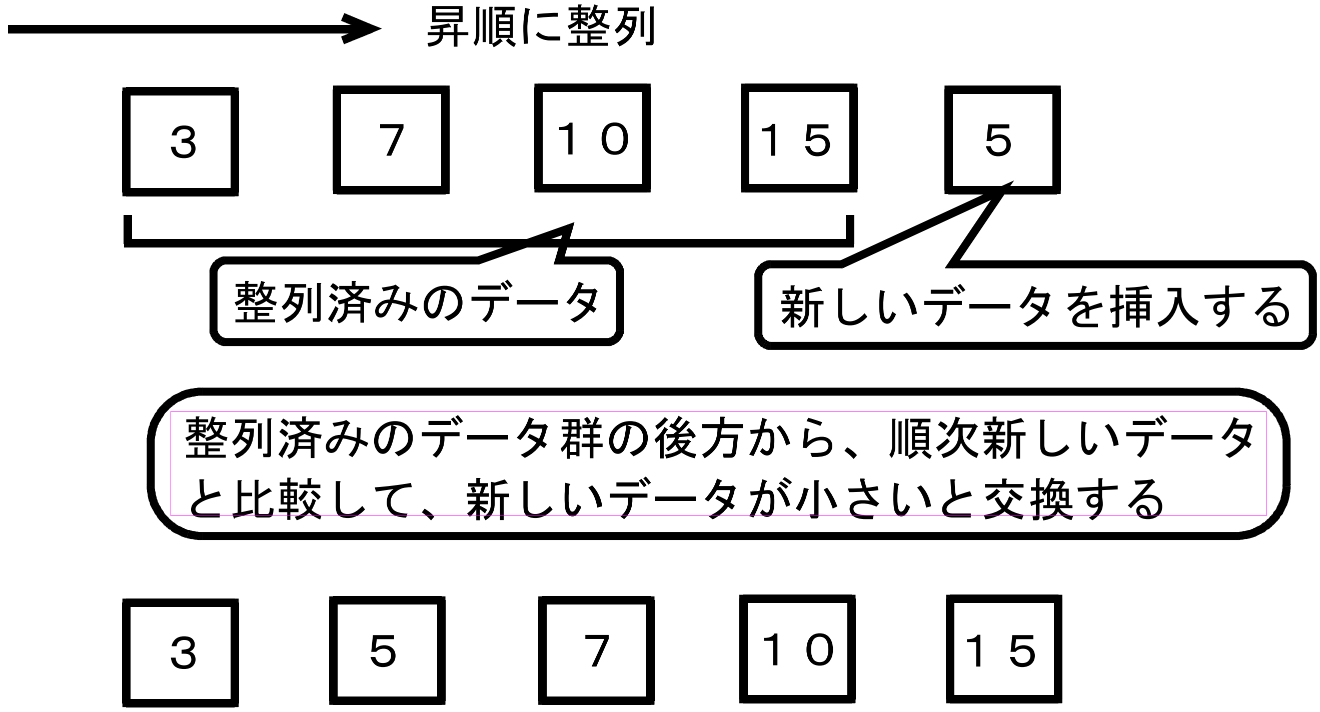
基本交換法の比較回数はn^2に比例する。

◆基本挿入法（挿入ソート）

◇基本挿入法

基本挿入法は、整列済みのデータ系列に対して、データを１件ずつ正規の位置に挿入し、その位置以降のデータを順次後方に移動させて、並べ替える。

新しいデータの挿入位置の決定は、列の後方から比較して順次前方に向かって調べる。



◇基本挿入法の手順

① 整列済みのデータの最後尾の要素の次に新しいデータを位置づける。

② 新しいデータと整列済みの最後尾のデータを比較し、大小関係が異なると、２個のデータを交換する。

③　交換すると、整列済みの次の前のデータと新しいデータを比較して、大小関係が異なると、２個のデータを入れ替える。

④　比較する２個のデータが異なる間、②～③の操作を繰り返す。

⑤　大小関係が正しくなると、その位置が新しいデータの挿入位置になる。

⑥　新しいデータがなくなるまで、①～⑤の操作を繰り返す。

新しいデータの挿入位置が決まると、それより後のデータの以前の位置よりも一つだけ後方に移動する。

◇基本挿入法の比較回数

基本挿入法の比較回数はn^2に比例する。

最大の比較回数は、新しく挿入するデータが整列済みのデータの先頭に挿入される場合である。

１回目の挿入データの比較回数は１回である。整列後のデータ数は２個になる。

２回目の挿入データの比較回数は２回である。整列後のデータ数は３個になる。

この操作を順次繰り返して、Ｎ－１目の挿入データのＮ－１回で、整列後のデータ数はＮ個となる。

比較回数の総和Ｓを求めると、次のようになる。

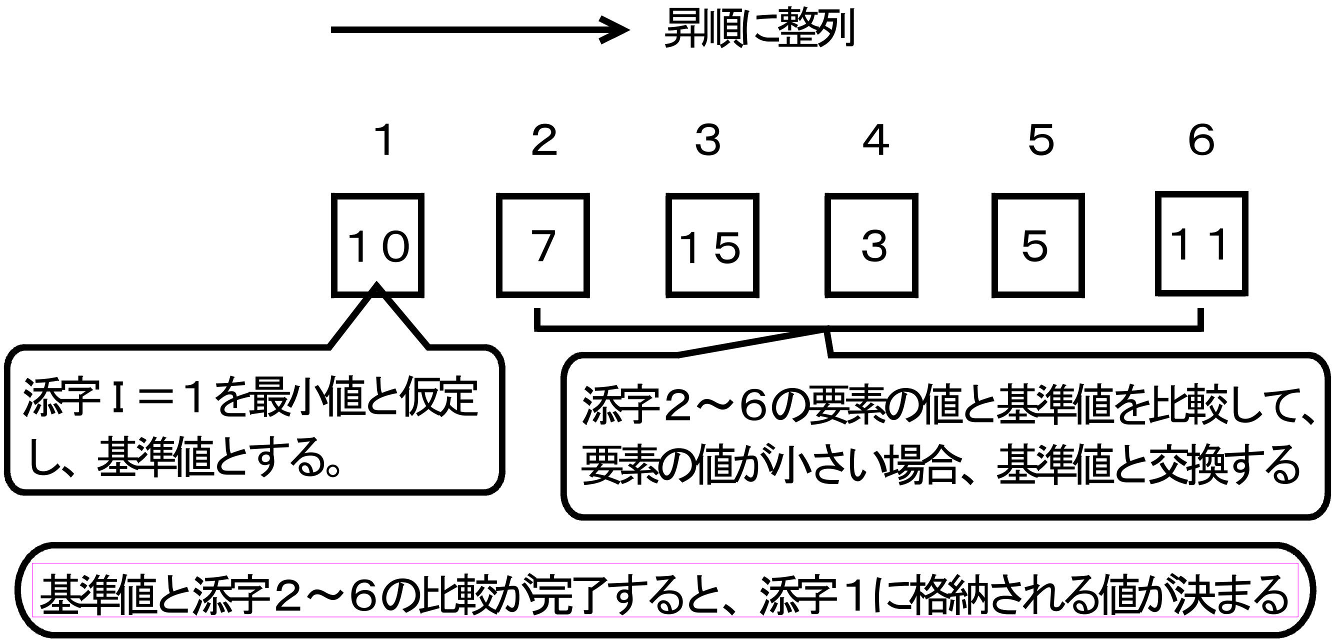
$$
\begin{aligned}
S &= 1 + 2 + 3 + \cdots + (N - 2) + (N - 1) \\
&= N \times (N - 1) / 2 \\
&= (N^2 - N) / 2
\end{aligned}
$$

◆基本選択法（選択ソート）

◇基本選択法

基本選択法は、隣接データの並べ替えを行わず、各回の比較で対象データの最大値（または最小値）を求め、最終格納位置のデータと交換する。

最も左の要素をデータ格納位置と決めると、残りのデータの最小値と格納位置のデータを比較し、格納位置のデータが大きい場合に最小のデータと交換する。



◇基本選択法の手順

① 配列の先頭の要素を決定対象の要素として選択する。

② 決定対象の要素とその他の要素のデータを順次比較して、大小関係が異なると、データを交換する。

③　配列の最後まで比較すると、１回目の比較を終了する。この比較で先頭の要素の位置が決まる。

④　配列の残りの要素の先頭の要素を決定対象に選択し、②、③の操作を繰り返す。この比較で配列の２番目の要素の位置が決まる。

⑤　順次、残りの配列の先頭の要素を決定対象に選択し、②、③の操作を繰り返す。

⑥　配列の最後の一つ前の要素の位置が決まると整列の操作が完了する。

②の比較後、毎回のデータの交換を行わず、小さいデータの要素番号を仮の格納場所に設定し、仮の格納場所に設定した最も小さい要素の値とその次のデータを比較し、大小関係が異なると、仮の格納場所に最も小さい値の要素番号を設定する。

③の配列の最後まで比較し、１回目の比較を終了すると、決定対象の要素と仮の格納場所の要素とを入れ替える。決定対象の要素に最も小さい値が設定される。

◇基本選択法の比較回数

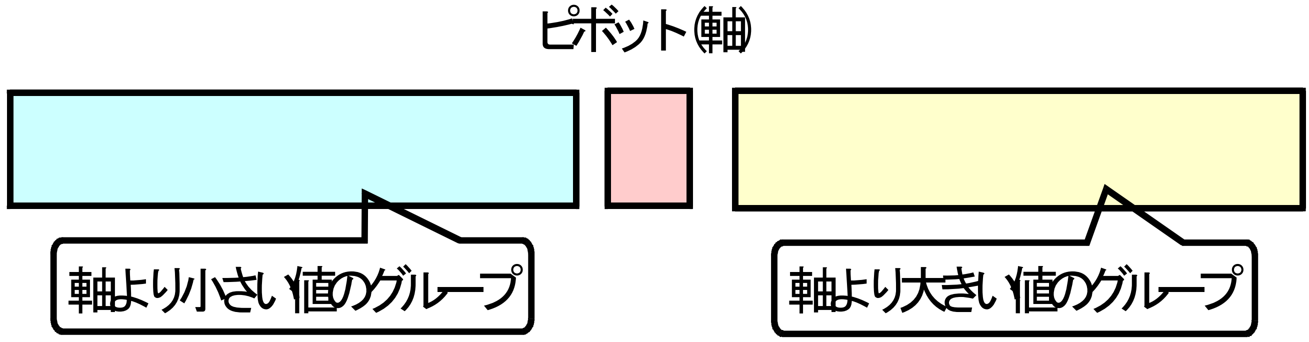
基本選択法の比較回数はｎ^{2}に比例する。

◆クィックソート

◇クィックソートとは

　適当にサンプリングして得られたデータＮに着目し、Ｎより小さいデータ系列とＮ以上のデータ系列に２分割する。このデータＮを軸（ピボット）という。

　データＮを軸として、その軸の値より小さい要素は軸より前方へ、大きな要素は軸より後方へ振り分ける。適当な長さの系列になるまで分割を繰り返し、それぞれの系列をソートすれば１つのソート済み系列となる。



　クィックソートは、分割の繰返しに再帰呼び出しを利用した高速の整列法である。

　ピボットの選び方には、中央値や３要素（右・中央・左）の中間値の考え方などがある。

◇クィックソートのアルゴリズム

　数列の左側から走査していく変数をｉ、右側から走査していく変数をｊとすると、最左端の基準値に対して、左部分列と右部分列の走査は次のようになる。

①　ｉを数列の最左端＋１、ｊを数列の右端に設定する。

②　数列を左から右に走査して、要素の値が基準値以上になる位置ｉを見つける。

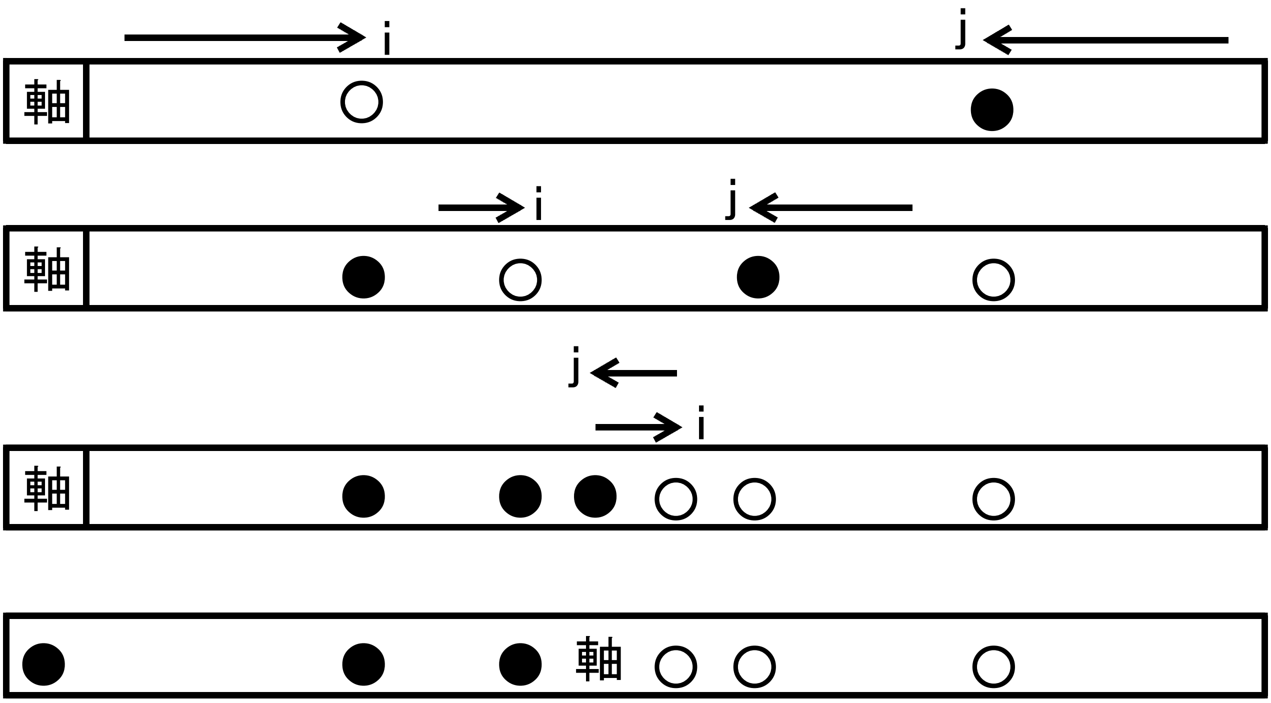
③　数列を右から左に走査して、要素の値が基準値より小さくなる位置ｊを見つける。

④　ｉ≧ｊならば、ループを抜け、⑥に進む。

⑤　ｉ＜ｊならば、ｉ項とｊ項を交換し、②に戻る。

⑥　基準値とｊ項を交換する。

　次回の対象になる左部分列は左端からｊ－１、右部分列はｊから右端の２つの範囲について同様の操作を繰り返す。



●軸より小さいもの　○軸より大きいもの

◇クィックソートの比較回数

データ数をｎとすると、平均比較回数、最大比較回数は次のようになる。

① 平均比較回数は$n\log_2 n$となる。

② 最大比較回数はn^2となる。

◇クィックソートの具体例

次に示す１０個のデータをクィックソートのアルゴリズを利用して整列する。

41 24 76 11 45 64 21 69 19 36

◇具体例の操作の手順

① 基準値を４１に設定し、左側の初期値を２４、右側の初期値を３６に設定する。

② 左側からの走査で基準値４１以上の値は７６、右側の走査から４１以下の値は３６となる。３６と７６を入れ替える。

41 24 36 11 45 64 21 69 19 76

③ 左側の走査で基準値以上は４５、右側の走査から４１以下の値は１９となる。４５と１９を交換する。

41 24 36 11 19 64 21 69 45 76

④ 左側の走査で基準値以上は６４、右側の走査から４１以下の値は２１となる。６４と２１を交換する。

1	2	3	4	5	6	7	8	9	10
41	24	36	11	19	21	64	69	45	76

⑤　左側の走査で基準値以上は６４、右側の走査から４１以下の値は２１となり、６４の要素番号は７、２１の要素番号は６となり、７＞６となる。

⑥　左側の基準値４１と２１を交換すると、

21	24	36	11	19	41	64	69	45	76

この操作によって、４１を基準値にして、左部分列と右部分列が求められたことになる。

次には、２１～４１の左部分列のグループと６４～７６の右部分列のグループの２つのグループに分けて、それぞれのグループの基準値を設定し、同様の処理を繰り返す。

最後に、１個または２個のグループが複数個でき、それぞれのグループ内の整列が完了すると、全体の整列も完了する。

各グループの操作は、すべて同じ手順で実行できるため、再帰呼び出しが利用される。

◆シェルソート

◇シェルソートとは

　データ列の中からＸ間隔ずつ離れたデータを取り出して、その部分列のデータを挿入ソートの考え方を利用して整列する。

　このＸ間隔のギャップを、始めは大きく、次第に小さくして最後に１にする。

　ギャップの決め方は、データ数の１／２、１／４　、…、１とする方式を用いる。

　例えば、データが８個ある場合、最初の飛びは８／２＝４とし、飛びの対のデータについて挿入ソートを使用する。

　次に、４／２＝２の飛びの対について挿入ソートを使用し、最後に２／２＝１の飛び、通常の挿入ソートを行う。

　挿入ソートは、常に隣の要素と比較・交換するため、実行速度が遅いという欠点があったが、それを改善し、順番の違いを早く修正することができる。

◇シェルソートの比較回数

　比較回数は$n^{1.5}$に比例する。

◆シェーカソート

◇シェーカソートとは

昇順に整列する場合、バブルソートは、常に左から右に比較していき、右端に最大値がくるように整列する。

シェーカソートは、左から右へ比較していき、右端に最大値を固定した後に、逆に、右から左へ比較していき、最小値を左端に固定する方式である。この操作を繰り返してデータを昇順に並べ替える方式である。

左から右に比較していく場合、データを交換した最後の位置を認識し、その位置を起点にして右から左への比較を開始する。右から左に比較する場合も、データを交換した最後の位置を認識し、その位置を起点として左から右への比較を開始する。

◇シェーカソートの具体例

最初は左から右に走査し、最後に交換した１３の位置を記憶する。

```
19  10  26  43  21  13     比較回数左から５回
↑                   ↑
10  19  26  43  21  13
10  19  26  43  21  13
10  19  26  43  21  13
10  19  26  21  43  13
10  19  26  21  13  43
                ↑
```

次に１３の位置を出発点にして、右から左に走査し、最後に交換した１９の位置を記憶する。

10　19　26　21　13　43　　比較回数右から４回
↑　　　　　　　　↑
10　19　26　13　21　43
10　19　13　26　21　43
10　13　19　26　21　43
　　　　↑　　　　↑

　その次の走査は、１９の位置と１３と２１を交換した２１の位置との間で同じ走査を繰り返し、最後に交換した位置を記憶する。

　走査の左の起点Ｌと右の起点ＲがＬ≧Ｒとなると、走査を終了する。

◆ヒープソート

◇ヒープソートとは

ヒープソートは、ソートすべきデータをヒープ構造の木を利用して行う方法である。

ヒープソートは、根の部分のデータを取り出した後、ヒープを再構成する作業を繰り返せば、ソート済みデータ系列が得られる。

◇ヒープソートの比較回数

平均比較回数も最大比較回数も$n \log_2 n$となる。

◆整列アルゴリズムと計算量

◇バブルソートの計算量

　バブルソートは比較回数は多いが、特別な作業領域を必要としない特徴がある。要素数をｎとすると、比較回数Ｍは次の式を用いて計算する。

$$
\begin{aligned}
M &= (n-1)+(n-2)+\cdots+2+1 \\
&= (n-1)\times\{(n-1)+1\}/2 \\
&= (n^2-n)/2 \\
&\fallingdotseq n^2/2\ (\text{回})
\end{aligned}
$$

　バブルソートの手順は左端から順番に隣接要素を比較する。左側の要素＞右側の要素の時、左右の要素を交換する。この一連の処理を繰り返して、最大の要素を右端に位置づける。

　次に残りの要素に対して、再度左端から比較を行い最大値を右端に設定する。以上の処理をｎ－１回繰り返すと昇順に整列する。

　この場合の計算量はデータ数をｎとすると、n^2に比例する。

　図の交換操作で２５の位置が決定す手順は次の通りとなる。

①　１５と２５を比較する。昇順であり、交換必要なし。

②　２５と１１を比較する。大小が逆で交換する。

③　２５と１８を比較する。大小が逆で交換する。

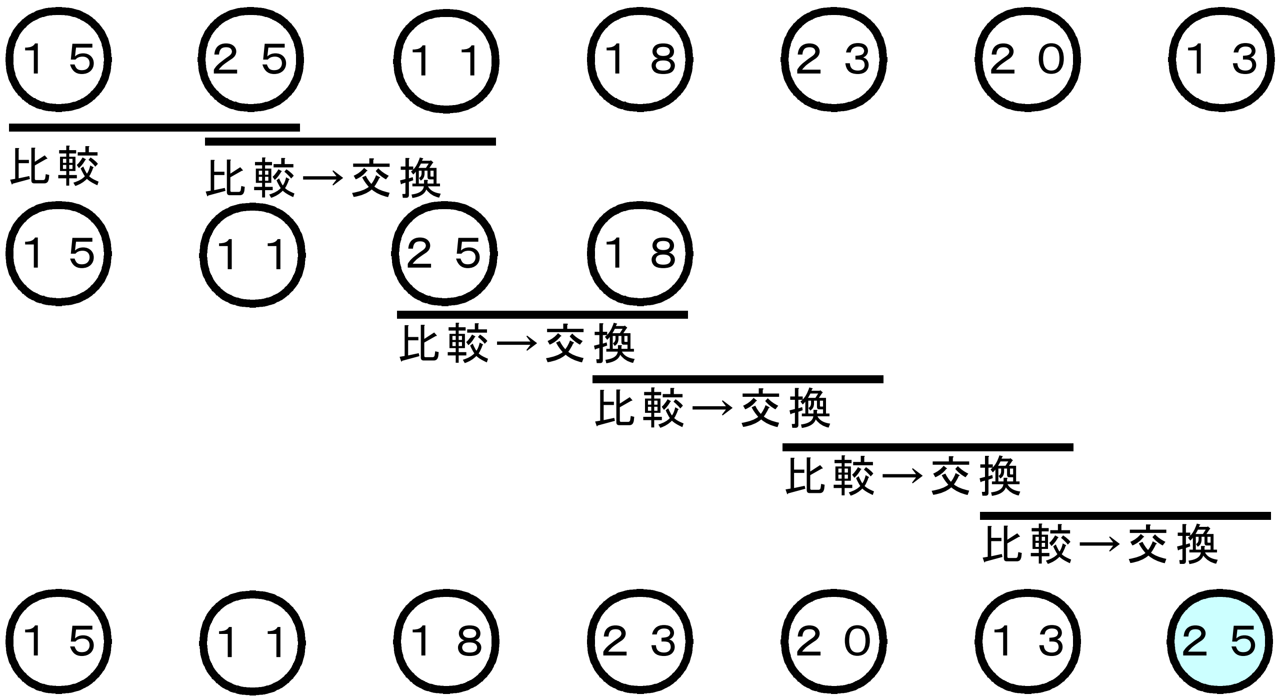
④　２５と２３を比較する。大小が逆で交換する。

⑤　２５と２０を比較する。大小が逆で交換する。

⑥　２５と１３を比較する。大小が逆で交換する。

⑦　２５の位置が決定する。

この場合の比較回数は６回である。



２５を除いた残りの６要素について、同様に比較・交換を行い、２３の位置を決定する。この２回目の比較回数は５回となる。

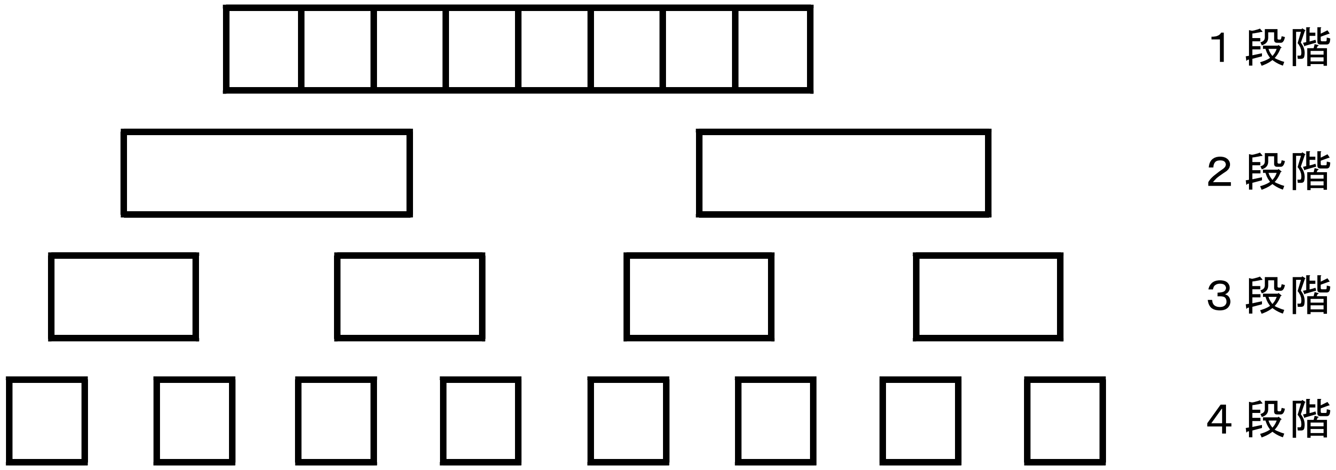
以下同様の操作を実行すると、比較回数は、４回、３回、２回、１回となる。

従って、全比較回数は

６＋５＋４＋３＋２＋１＝２１（回）

となる。

◇クィックソートの計算量



整列対象の要素数をｎとし、ある基準値を設定し、その基準値より大きい要素のグループと小さい要素のグループに分ける。

次の段階で、２つのグループにそれぞれの基準値を設定し、それぞれの基準値より大きい要素のグループを２グループ、小さい要素のグループを２グループ作成する。

次の段階で、４グループに対して同じ操作を繰り返す。各グループの要素数が１になるまで繰り返す。

１回目の比較回数はｎ回であり、各グループの要素が１になるまでの段階数ｋは次のようにして求める。

段階数をｋとすると、データ数ｎとの間には、次式が成立する。

$$2^k \leqq n < 2^{k+1}$$

２を底とする対数をとると

$$\log_2 2^k \leqq \log_2 n < \log_2 2^{k+1}$$

$$k \leqq \log_2 n < k+1$$

故に、段階数は　$k = [\log_2 n]$となる。

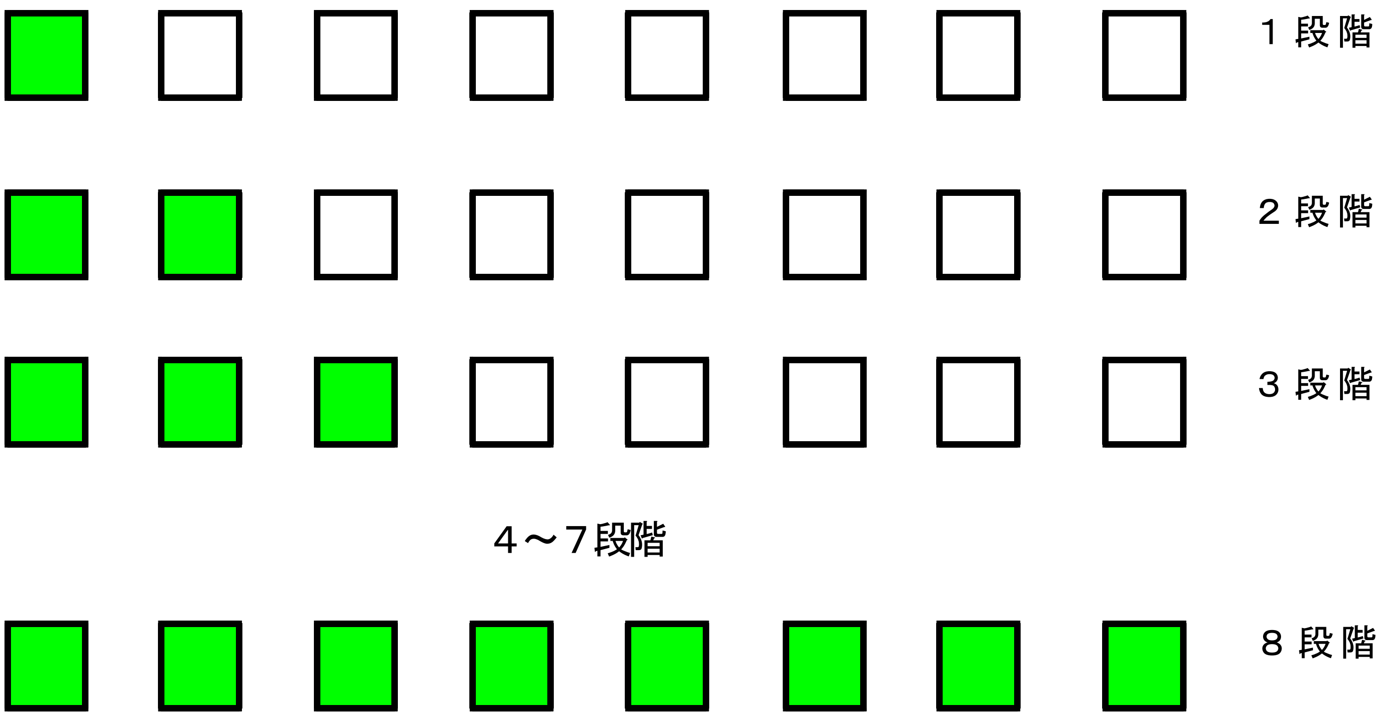
　全比較回数は、各段階での比較数ｎを乗じて

　　　　　ｎ×［ｌｏｇ$_{2}$ｎ］

となる。

　比較数が最大となるのは、各段階で基準値による振り分けが片側に偏る場合で、この場合の段階数はｎとなり、比較回数の合計はｎ^{2}になる。

　配列の最左端に基準値（緑色の要素）があり、その他の要素が、すべて、基準値より右側に位置する場合について考え、２段階目以降も同じ現象が発生すると、次に示す段階数が生じることになる。この現象が段階数が最も多くなる場合である。

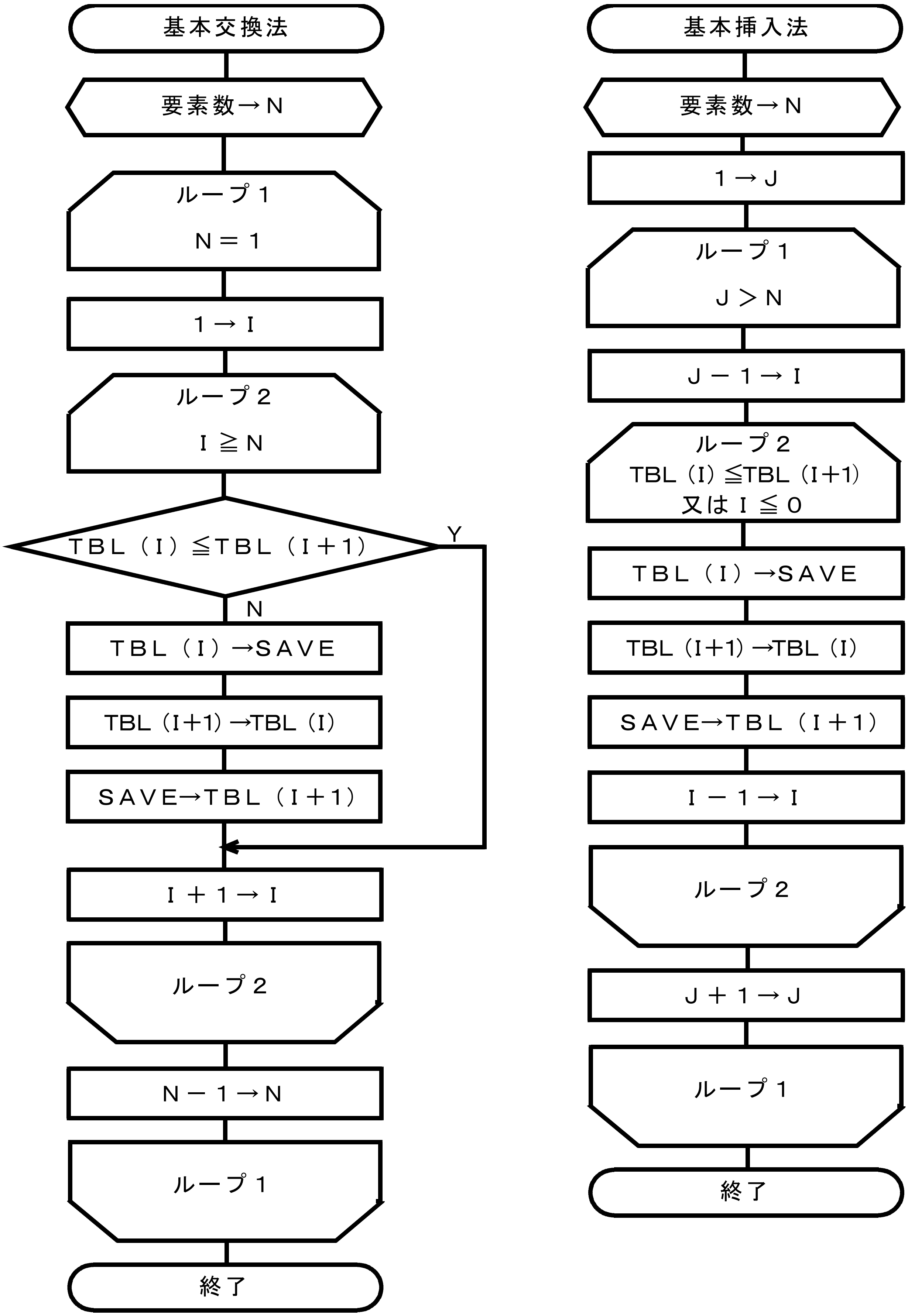


　クイックソートの計算量は、データ数をｎとすると、

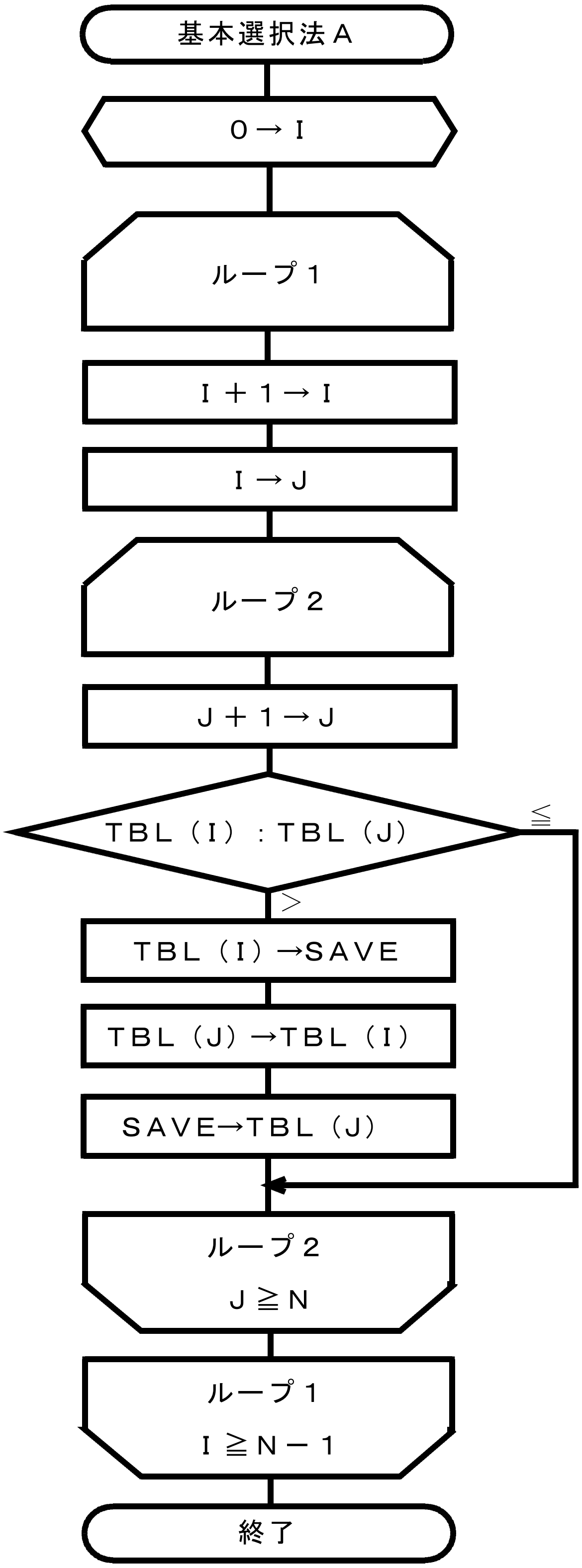
　　平均比較回数でｎ×［ｌｏｇ$_{2}$ｎ］
　　最大比較回数でｎ^{2}

となる。

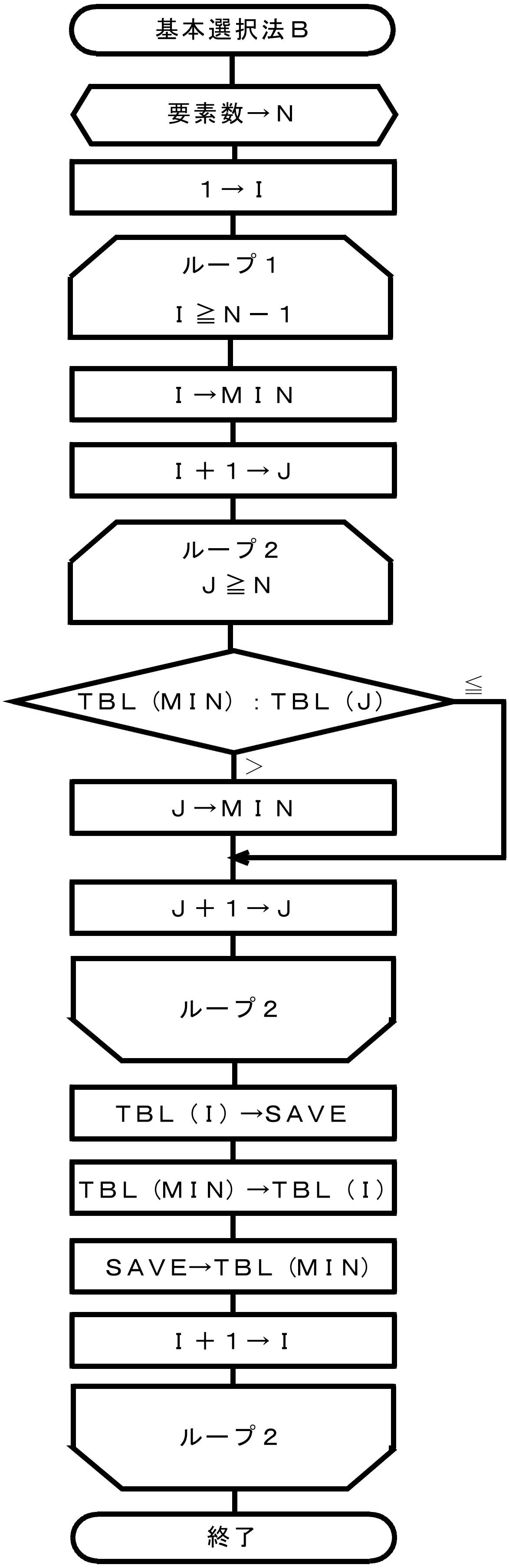
◇基本交換法・基本挿入法の流れ図



◇基本選択法の流れ図



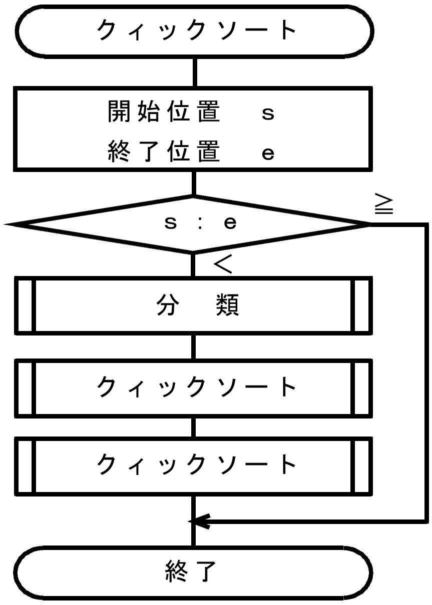
基本選択法Ａ
０→Ｉ
ループ１
Ｉ＋１→Ｉ
Ｉ→Ｊ
ループ２
Ｊ＋１→Ｊ
ＴＢＬ（Ｉ）：ＴＢＬ（Ｊ）
≦
＞
ＴＢＬ（Ｉ）→ＳＡＶＥ
ＴＢＬ（Ｊ）→ＴＢＬ（Ｉ）
ＳＡＶＥ→ＴＢＬ（Ｊ）
ループ２
Ｊ≧Ｎ
ループ１
Ｉ≧Ｎ－１
終了

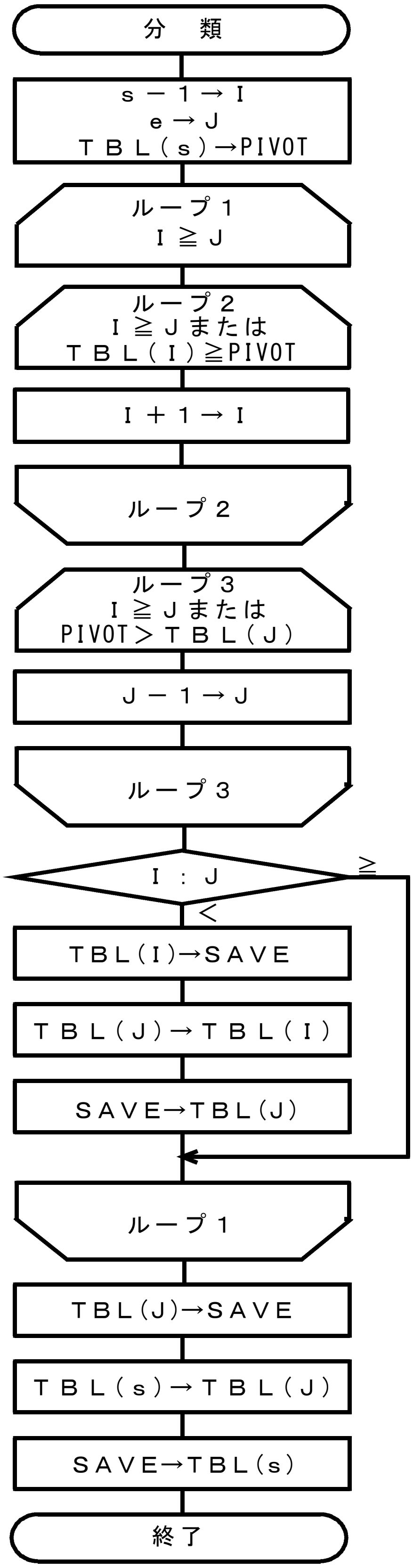


基本選択法Ｂ
要素数→Ｎ
１→Ｉ
ループ１
Ｉ≧Ｎ－１
Ｉ→ＭＩＮ
Ｉ＋１→Ｊ
ループ２
Ｊ≧Ｎ
ＴＢＬ（ＭＩＮ）：ＴＢＬ（Ｊ）
≦
＞
Ｊ→ＭＩＮ
Ｊ＋１→Ｊ
ループ２
ＴＢＬ（Ｉ）→ＳＡＶＥ
ＴＢＬ（ＭＩＮ）→ＴＢＬ（Ｉ）
ＳＡＶＥ→ＴＢＬ（ＭＩＮ）
Ｉ＋１→Ｉ
ループ２
終了

◇クイックソートの流れ図

クィックソートの流れ図





クィックソートの手順

① 開始位置ｓの要素をピボットにする。
② 左から走査する添字をＩ、右から走査する添字をＪとする。
③ 左からの走査でＩ＋１→Ｉを繰返し、ＴＢＬ（Ｉ）≧PIVOT になると停止する。
④ 右からの走査でＪ−１→Ｊを繰返し、ＴＢＬ（Ｊ）＜PIVOT になると停止する。
⑤ ＴＢＬ（Ｉ）とＴＢＬ（Ｊ）の値を交換する
⑥ Ｉ≧Ｊになるまで、③～⑤を繰り返す。
⑦ Ｉ≧Ｊになると、ＴＢＬ（ｓ）とＴＢＬ（Ｊ）を交換する。
⑧ 分割された２つの範囲について、再帰の考え方を使用して①～⑦を繰り返す。

サブルーチンの引数

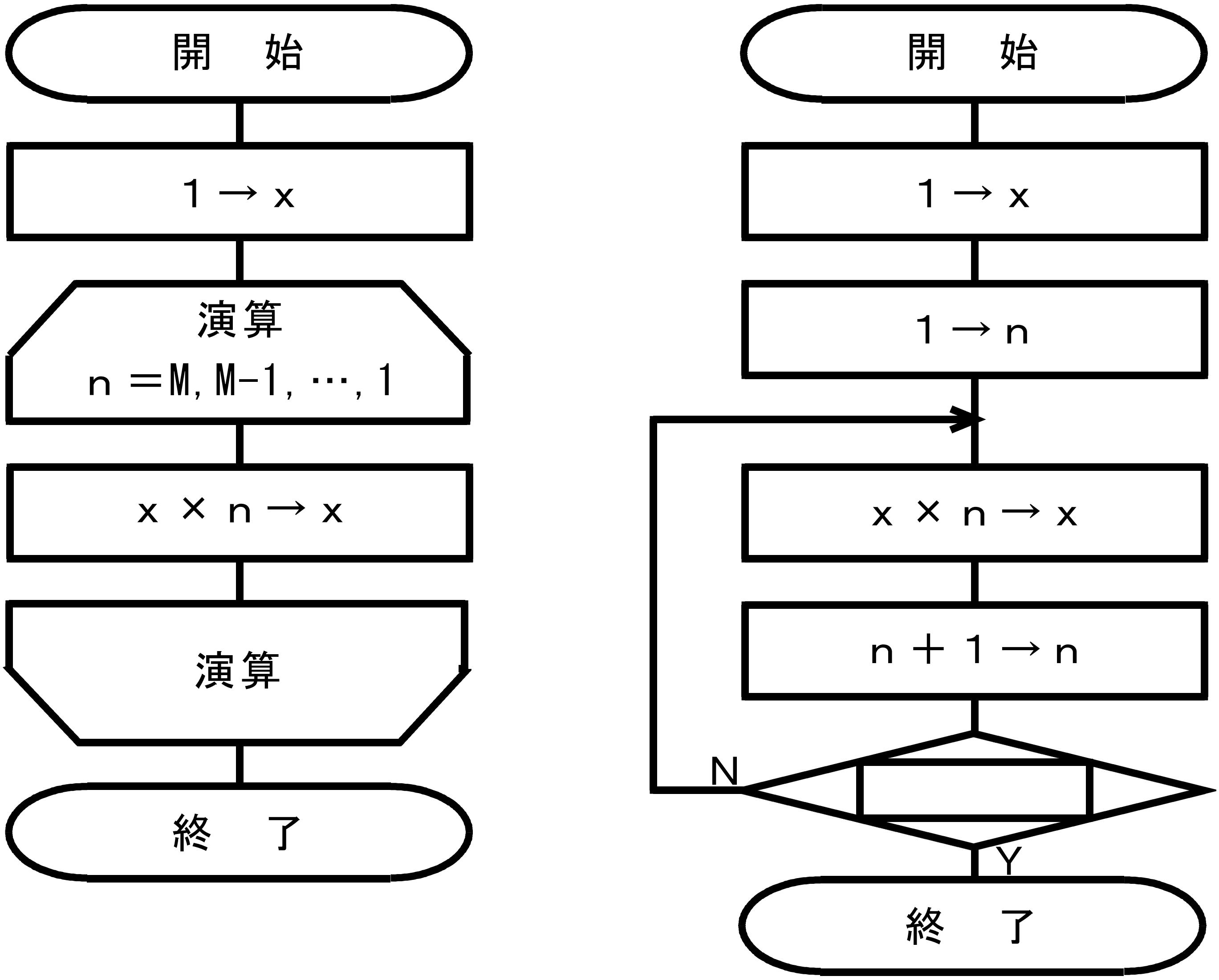
◆ クィックソート１でサブルーチンを呼び出す場合の引数は、ｓ、Ｊ−１となる。

◆ クィックソート２でサブルーチンを呼び出す場合の引数は、Ｊ、ｅとなる。

アルゴリズム問題１

問１問題

次の二つの流れ図に示したアルゴリズムを実行したとき、結果のｘの値が同じになるようにしたい。判断記号の［　　　］に入れる条件として正しいものはどれか。



ア　ｎ＞Ｍ
イ　ｎ＞Ｍ＋１
ウ　ｎ＞Ｍ－１
エ　ｎ＜Ｍ

◇問１解説

流れ図の問題である。

左側のフローチャートの結果は、Ｍ×（Ｍ－１）×…×２×１となる。

右側のフローチャートは、ｎのインクリメントがｘ×ｎの実行後であるから、Ｍまで乗じて、Ｍ＋１になったときに終了すればよい。従って、終了条件はｎ＞Ｍとなる。求める答えはアとなる。

◇問１解答　ア

問２問題

　次の探索方法のうちで番兵が有効なものはどれか。

ア　２分探索
イ　線形探索
ウ　ハッシュ探索
エ　幅優先探索

◇問２解説

　線形探索の番兵に関する問題である。

　番兵が有効なのは線形探索のように、並びの先頭から順次探索していく場合である。

　番兵は探し求める値を配列の最後に付加して、配列の大きさＮの範囲内に求める値が見つかれば探索成功、Ｎ＋１で見つかると探索不成功の判定をする線形探索法である。従って、番兵が有効となるのは線形探索である。求める答えはイとなる。

◇問２解答　イ

問３問題

ｎ個の要素を持つ配列中の値と探索すべきデータＸを順次比較し、配列中の値にデータＸが存在した場合、“有”を表示する。このとき、添字ｎ＋１の場所に探索すべきＸを入れておく。

添字	１	２	…	ｉ	…	ｎ	ｎ＋１
値	a_1	a_2		a_i		a_n	Ｘ

この線形探索アルゴリズムの$\boxed{}$に入れるべき適切な条件はどれか。

ステップ１　添字ｉに１を入れる。
ステップ２　$\boxed{}$であればステップ５へとぶ。
ステップ３　添字ｉに１を加算する。
ステップ４　ステップ２へとぶ。
ステップ５　添字ｉがｎ以下であれば“有”を表示する。
ステップ６　終了

ア　$i \geqq n$
イ　$i \neq n$
ウ　$X = a_i$
エ　$X \neq a_i$

◇問３解説

線形探索の番兵に関する問題である。

番兵は、探し求める値を配列の最後に付加して、配列の大きさＮの範囲内に求める値が見つかれば探索成功、Ｎ＋１で見つかると探索不成功の判定をする線形探索法である。

　番兵を利用した線形探索のアルゴリズムで、繰返しからの脱出条件は

　　　配列の値＝X

即ちX＝a_iである。求める答えはウとなる。

◇問３解答　ウ

問４問題

あらかじめソートされていることが必須条件である探索方法はどれか。

ア　Ｂ木探索
イ　線形探索
ウ　２分探索
エ　２分探索木

◇問４解説

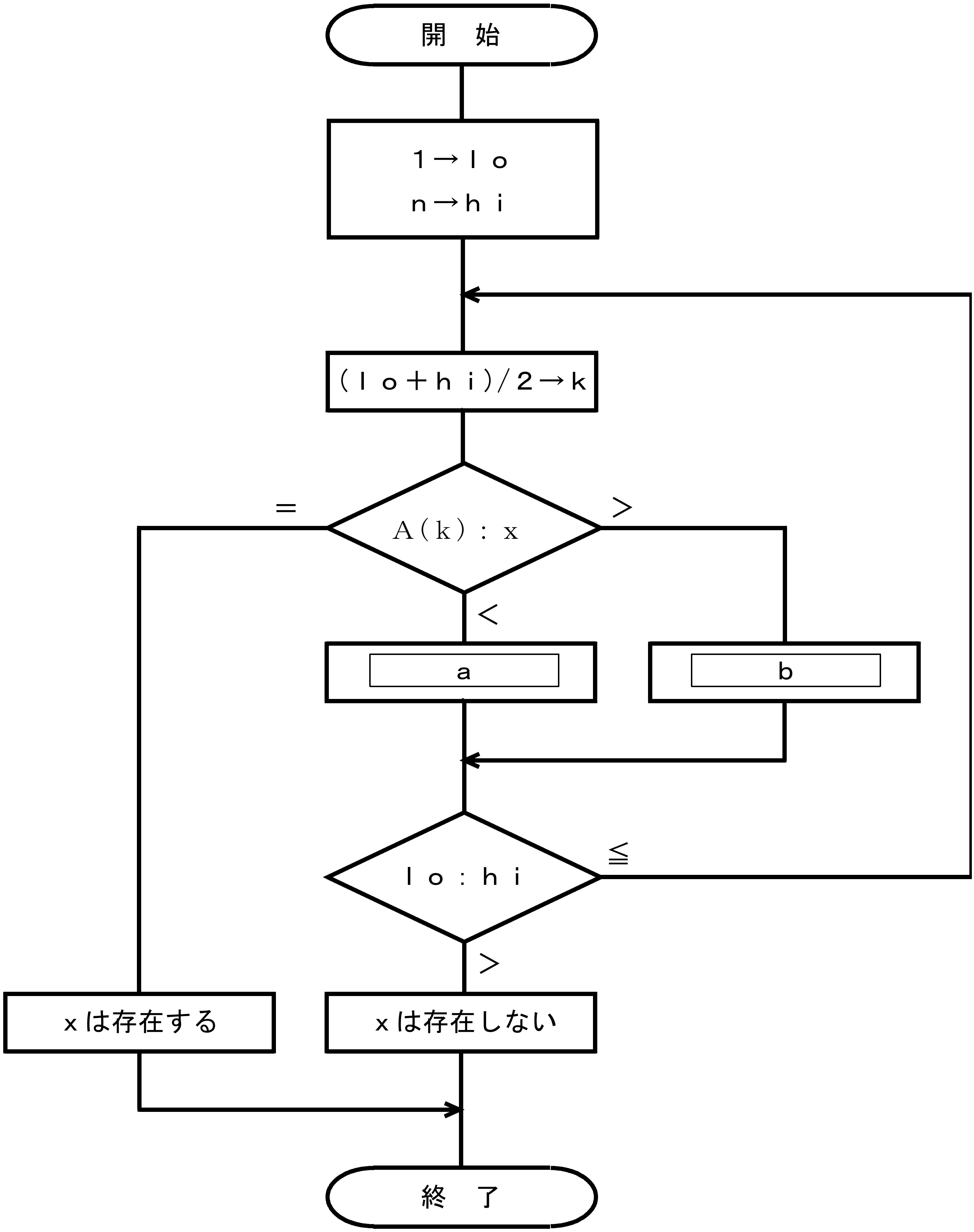
探索方法に関する問題である。

探索のための必須条件としてのソートとデータ構造として整列の条件を満たしているものとの区別が重要である。

アのＢ木探索、エの２分探索木は、定義の段階で既に整列済みであり、ソートを必要としない。
イの線形探索は、配列の先頭または後方から順次探索する方法であり、ソートを必要としない。
ウの２分探索は、中央の値と探索データを比較して、その大小関係によって、探索する範囲を１／２にして、新たな中央の値を求め、比較を繰り返す方法であり、探索する前に昇順または降順に整列しておかないと探索することができない。あらかじめソートが必須条件は２分探索である。求める答えはウとなる。

◇問４解答　ウ

問５問題



昇順に整列されたｎ個のデータが格納されている配列Ａがある。

流れ図は、配列Ａからデータｘを２分探索法を用いて探し出す処理である。ａ、ｂに入る操作の正しい組合せはどれか。

ここで、除算の結果は小数点以下切捨てとする。

	a	b
ア	ｋ＋１→ｈｉ	ｋ－１→ｌｏ
イ	ｋ－１→ｈｉ	ｋ＋１→ｌｏ
ウ	ｋ＋１→ｌｏ	ｋ－１→ｈｉ
エ	ｋ－１→ｌｏ	ｋ＋１→ｈｉ

◇問５解説

２分探索法の流れ図に関する問題である。

ａは探索すべき値ｘが比較する配列の値より大きい場合で、次の探索範囲はｋ＋１～ｈ$_{i}$の範囲になる。この場合は、上限は変化しないが、下限ｌ$_{o}$＝ｋ＋１となる。

ｂは探索すべき値ｘが比較する配列の値より小さい場合で、次の探索範囲はｌ～ｋ－１の範囲となる。この場合は、下限は変化しないが、上限はｈ$_{i}$＝ｋ－１となる。
求める答えはウとなる。

◇問５解答　ウ

問６問題

　整列済みのデータを対象とした二分探索に関する次の記述の中で正しいものはどれか。

ア　データが昇順に整列されている場合、キー値の小さいデータの方がキー値の大きいデータより、少ない比較回数で探索できる。
イ　検索するキーがデータの中にないと。無限ループに陥る。
ウ　データの要素数が２倍になると、キー値の最大比較回数は４倍になる。
エ　データ中に同じキー値をもつ要素が複数個ある場合、検索の結果発見できるのは、その中の特定の一つだけである。

◇問６解説

　２分探索法の特徴に関する問題である。

　アは、検索するキー値が小さい方になるか大きい方になるかは同じ確率で発生する。従って、キー値の小さいデータの値がキー値の大きいデータよりも少ない比較回数で探索できるとは限らない。
　イの検索するデータが存在しない場合は、最後に探索対象の範囲内の要素数が０となり、二分探索が終了する。
　ウの要素数が２倍になっても探索回数は１回増加する程度である。
　エは、二分探索で比較されるのは、対象の範囲の中央の要素であるから、同じ内容の要素が複数存在しても、中央の要素になった一つが探索されるのみである。求める答えはエである。

◇問６解答　エ

問７問題

次の記述の$\boxed{}$の中に入れる適当な語句はどれか。

昇順に整列済みのｎ個のデータに対して二分探索を行う場合の探索終了条件は、探索成功または$\boxed{}$となっている。

ア　小さいほうの要素番号≧大きいほうの要素番号
イ　小さいほうの要素番号＞大きいほうの要素番号
ウ　小さいほうの要素番号＞ｎ、または、大きいほうの要素番号＜１
エ　小さいほうの要素番号≧ｎ、または、大きいほうの要素番号≦１

◇問７解説

２分探索の探索終了条件に関する問題である。

２分探索法の終了条件

①　探索データが見つかった場合
②　整列済みの配列に対して、探索範囲の上限の添字をＨ、下限の添字をＬとすると、Ｌ＞Ｈになれば探索を終了する。

アの等号は探索が可能であり、終了条件にはならない。
ウ、エは探索範囲が変化するため最初の両端の１、ｎでは判定できない。
求める答えはイとなる。

◇問７解答　イ

問８問題

　次の状態で記録されたファイルで、２分探索に最も適しているものはどれか。

ア　磁気ディスク上に索引順編成で記録したファイル
イ　磁気テープ上にレコードをハッシュ法によって記録したファイル
ウ　主記憶上にレコードをキー順に記録したファイル
エ　主記憶上にレコードを線形リストで記録したファイル

◇問８解説

　２分探索に関する問題である。

　２分探索の条件はキー値が昇順または降順に整列されていることである。

　アの索引順編成はデータは入力順に並んでおり索引を利用して目的のデータを効率よく探索するものであるから、必ずしも２分探索には適さない。
　イのハッシュ法によって記録されたデータはハッシュ関数の計算結果で格納場所が決められており、２分探索可能なデータの並びになっていない。
　ウはキー順に記録されたファイルであり、昇順または降順に整列したキーを利用して２分探索が可能である。求める答えはウである。
　エの線形リストで記録されたデータは昇順または降順に並んでいるとは限らない。従って、２分探索には適さない。

◇問８解答　ウ

問９問題

ハッシュ法の説明として，適切なものはどれか。

ア　関数を用いてレコードのキー値からレコードの格納アドレスを求めることによってアクセスする方法
イ　それぞれのレコードに格納されている次のレコードの格納アドレスを用いることによってアクセスする方法
ウ　レコードのキー値とレコードの格納アドレスの対応表を使ってアクセスする方法
エ　レコードのキー値をレコードの格納アドレスとして直接アクセスする方法

◇問９解説

ハッシュ法に関する問題である。

ハッシュ法は、ハッシュ関数を使用して、データの探索キーの値をデータの格納アドレスに変換することである。データの参照を高速で行う。

ハッシュ法では、異なるキーから同じハッシュ値が求まる場合があり、これを衝突という。衝突した場合の対策として、チェーン法とオープンアドレス法がある。

チェーン法は同じハッシュ値を持つデータをポインタを利用してリストでつなぐ方法である。オープンアドレス法は、元のハッシュ値＋１で再ハッシュを行う方法である。

ハッシュ法はアドレスを計算で求めるため、計算量は通常Ｏ(1)である。従って、表探索に利用する場合も衝突が発生しなければ１回の計算で格納場所を決めることができる。

アの関数を用いてレコードのキー値からレコードの格納アドレスを求めるアクセス法はハッシュ法に関する内容である。求める答えはアとなる。

イのレコードに格納されている次のレコードの格納アドレスを用いる方法はリスト構造のデータのアクセスに使用する方法である。

ウのレコードのキー値と格納アドレスの対応表を使用する方法は索引編成の索引を用いるアクセス方式である。

エのレコードのキー値を格納アドレスとして使用する方法は直接編成の実アドレス指定方式である。

◇問９解答　ア

問10問題

　１６進数で表される９個のデータ１Ａ，３５，３Ｂ，５４，８Ｅ，ＡＩ，ＡＦ，Ｂ２，Ｂ３を順にハッシュ表に入れる。ハッシュ値をハッシュ関数ｆ（データ）＝ mod(データ, 8)で求めたとき，最初に衝突が起こる(既に表にあるデータと等しいハッシュ値になる)のはどのデータか。
　ここで，mod(a, 6)はａをｂで割った余りを表す。

ア　５４
イ　Ａ１
ウ　Ｂ２
エ　Ｂ３

◇問10解説

　シノニムデータに関する問題である。

　与えられた９個のデータの処理内容の表を作成する。

　次の手順で求める。

①　９個の１６進数のデータを１０進数に変換する。

②　ハッシュ関数を使用してハッシュ値を求める。

③　最初にハッシュ値が一致するデータを求める。

16進数	1A	35	3B	54	8E	A1	AF	B2	B3
10進数	26	53	59	84	142	161	175	178	179
ハッシュ値	2	5	3	4	6	1	7	2	3

　従って、最初にハッシュ値が一致するのはＢ２の２である。求める答えはウとなる。

（別解）
　この問題ではハッシュ値を次のようにした求めることができる。

　１６進数を除数８で割る場合、１６進数の２桁目より上位は必ず割り切れて余りは０となる。従って、ハッシュ値は１６進数の１桁の値を８で割った余りである。

　１桁目の１６進数が０～７までの場合はその値が余りになり、８～１５までの場合は８を減じた数値になる。

◇問10解答　ウ

問11問題

ハッシュ法を用いて探索を行う場合、複数の検索要素に対するハッシュ値が等しくなるのは次のどれか。

ア　オーバフロー
イ　マッチング
ウ　コリジョン
エ　パディング

◇問11解説

ハッシュ法の衝突に関する問題である。

アのオーバフローは、溢れることで、算術演算の結果、数の表現のために与えられた語長を越える現象を言う。
イのマッチングは、突き合わせで、２つのファイルを照合してレコードを抽出したり、マスタファイルを更新したりする操作である。
ウのコリジョンは、衝突で、直接編成ファイルのアドレス変換において、異なるキーが同じアドレスに変換されることである。求める答えはウである。
エのパディングは、包含で、一つのメソッドが別のメソッドを参照して利用する関係をいう。

◇問11解答　ウ

問12問題

５けたの数字（$a_1 a_2 a_3 a_4 a_5$）をハッシュ法を用いて配列に格納したい。

ハッシュ関数をmod（$a_1+a_2+a_3+a_4+a_5$，１３）とし、求めたハッシュ値に対応する位置の配列要素に格納する場合、５４３２１は次の配列のどの位置に入るか。

ここで、mod（Ｘ，１３）の値は、Ｘを１３で割った余りとする。

位置	配列
0	
1	
2	
	⋮
１１	
１２	

ア　1
イ　2
ウ　7
エ　１１

◇問12解説

ハッシュ関数を使用してアドレスを計算する問題である。

ハッシュ関数の内容は、５桁の数字を桁別に分離してその和を求め、その結果を１３で割って余りを求める算出法を示している。

ハッシュ値を求めると

５＋４＋３＋２＋１＝１５
mod（１５，１３）＝２

求める答えはイとなる。

◇問12解答　イ

問13問題

　キー値の分布が１～１，０００，０００の範囲で一様ランダムであるデータ５個を、大きさ１０のハッシュ表に登録する場合、衝突の起こる確率はおよそ幾らか。

　ここで、ハッシュ値はキー値をハッシュ表の大きさで割った余りを用いる。

ア　０．２
イ　０．５
ウ　０．７
エ　０．９

◇問13解説

　ハッシュ法の衝突に関する問題である。

　５個の抽出されたデータを大きさ１０のハッシュ表に重複を許して登録する場合の「場合の数」と重複を許さないで登録する場合の「場合の数」を求め、前者に対する後者の割合、すなわち、衝突が発生しない割合を求め、１から衝突しない割合を減ずると衝突する割合を求めることができる。

①　衝突が発生する場合を含めた５個のデータのハッシュ表に登録される場合の数は$１０^{５}$通りである。

②　衝突しない場合の場合の数は、

　　　１０×９×８×７×６＝３０２４０

　通りになる。

③　衝突しない確率は

　　　３０２４０／１０００００＝０．３

④　衝突する確率は

　　　１－０．３＝０．７

　となる。

　求める答えはウとなる。

<u>◇問13解答　ウ</u>

問14問題

キーｘのハッシュ関数としてｈ（ｘ）＝mod（ｘ，９７）を用いるとき，キー１０９４とハッシュ値が一致するものは，キー１～１０００の中に幾つあるか。

ここで，mod（ｘ，９７）はｘを９７で割った余りを表す。

ア　９
イ　１０
ウ　１１
エ　１２

◇問14解説

ハッシュ法に関する問題である。

キー１０９４のハッシュ値を求めると、

mod（１０９４，９７）＝２７

キー１～１０００の間で、ハッシュ値が２７になるのは、９７×０＋２７、９７×１＋２７、…、９７×１０＋２７の１１個となる。求める答えはウとなる。

◇問14解答　ウ

問15問題

　００００～４９９９のアドレスをもつハッシュ表があり，レコードのキー値からアドレスに変換するアルゴリズムとして基数変換法を用いる。キー値が５５５５０のときのアドレスはどれか。

　ここで，基数変換法ではキー値を１１進数と見なし，１０進数に変換した後，下４けたに対して０．５を乗じた結果（小数点以下は切捨て）をレコードのアドレスとする。

ア　０２６０
イ　２５２５
ウ　２７７５
エ　４４０５

◇問15解説

　ハッシュ法に関する問題である。

　ハッシュ関数に基数変換法を用いる問題である。
　基数変換法を用いて、１１進数のキー値５５５５０を１０進数に変換する。

$$
\begin{aligned}
&5\times(11^{4}+11^{3}+11^{2}+11)+0\\
&=5\times(14641+1331+121+11)\\
&=5\times16105\\
&=70520
\end{aligned}
$$

　求めるアドレスは０５２０×０．５＝０２６０
　求める答えはアとなる。

◇問15解答　ア

問16問題

　データ列の隣り合う要素の値を比較し，小さい方が右にあれば交換する。この操作をデータ列の左端から右端まで繰り返す処理を１回のパスとする。次のデータ列でパスを２回繰り返した後のデータ列の内容を示しているものはどれか。

5	4	1	3	6	2

ア

1	3	2	4	5	6

イ

1	3	4	2	5	6

ウ

4	1	5	3	2	6

エ

4	1	5	3	6	2

◇問16解説

　バブルソートの並べ替えに関する問題である。

　並べ替えは次のようになる。

① 　１回目の並べ替え

5　4　1　3　6　2
4　5　1　3　6　2
4　1　5　3　6　2
4　1　3　5　6　2
4　1　3　5　2　6

② 　２回目の並べ替え

4　1　3　5　2　6

１　<u>４　３</u>　５　２　６
１　３　４　<u>５　２</u>　６
１　３　４　２　５　６

２回目の並べ替えを完了した時点での並びは

１　３　４　２　５　６

の順であり、求める答えはイとなる。

<u>◇問16解答　イ</u>

問17問題

データの整列と併合に関する次の記述中の$\boxed{}$に入れるべき適切な語句の組合せはどれか。

キーの値の小さいものから大きなものへデータを並べることを、$\boxed{\text{a}}$に$\boxed{\text{b}}$するという。対象とするデータ列が補助記憶装置にある場合、この操作を$\boxed{\text{c}}$と呼ぶ。

また、一定の順序に$\boxed{\text{b}}$された二つ以上のファイルを統合することを$\boxed{\text{d}}$という。

	a	b	c	d
ア	降順	整列	外部整列	併合
イ	降順	併合	内部併合	整列
ウ	昇順	整列	外部整列	併合
エ	昇順	併合	内部併合	整列

◇問17解説

整列、併合に関する問題である。

補助記憶装置を用いて並べ替えを行う場合を外部整列とか外部分類という。これに対して主記憶装置内だけで整列することを内部整列または内部分類という。　整列済みの複数のファイルをキー項目に従って順序を崩さずに１本のファイルに統合することを併合という。

ａは昇順、ｂは整列、ｃは外部整列、ｄは併合で、求める答えはウである。

◇問17解答　ウ

問18問題

データの整列方法に関する記述のうち、正しいものはどれか。

ア　クィックソートは、ある間隔で要素を取り出した部分列を整列し、更に間隔をつめた部分列を取り出して整列する方法である。
イ　シェルソートは、隣り合う要素を比較して、大小の順が逆であれば、それらの要素を入れ替えるという操作を繰り返して行う方法である。
ウ　バブルソートは、中間的な基準値を決めて、それより大きな値の要素を集めた区分と小さな値の要素を集めた区分とに振り分ける。次にそれぞれの区分の中で同様な処理を繰り返す方法である。
エ　ヒープソートは、未整列の部分を部分木で表し、そこから最大値又は最小値を取り出して既整列の部分に移す。この操作を繰り返して、未整列部分を縮めていく方法である。

◇問18解説

データの整列に関する問題である。

バブルソート(基本交換法)は、配列の隣り合ったデータを比較して、順序が違っていると並び替えていく方法であり、配列の先頭と次ぎのデータの比較から始め、配列の最後まで来たら１回目の比較を終了する。１回の操作で最後の要素の位置が決まる。２回目以降も同様の比較を行い、各回の操作で最後の位置が決まる。

クイックソートは、軸となるデータに着目し、そのデータより大きい系列と小さい系列に分類する。新しくできた二つの系列に対してそれぞれ軸となるデータを設定し、大きい系列と小さい系列に分額する。この操作を繰り返しながら対象のデータを昇順または降順に並べ替える方法である。操作の展開には再帰の考え方

が用いられる。

　シェルソートは、挿入ソートを改良したものであり、データ列の中からX間隔ずつ離れたデータを取り出して、その部分列のデータを挿入ソートの考え方を利用して整列する。

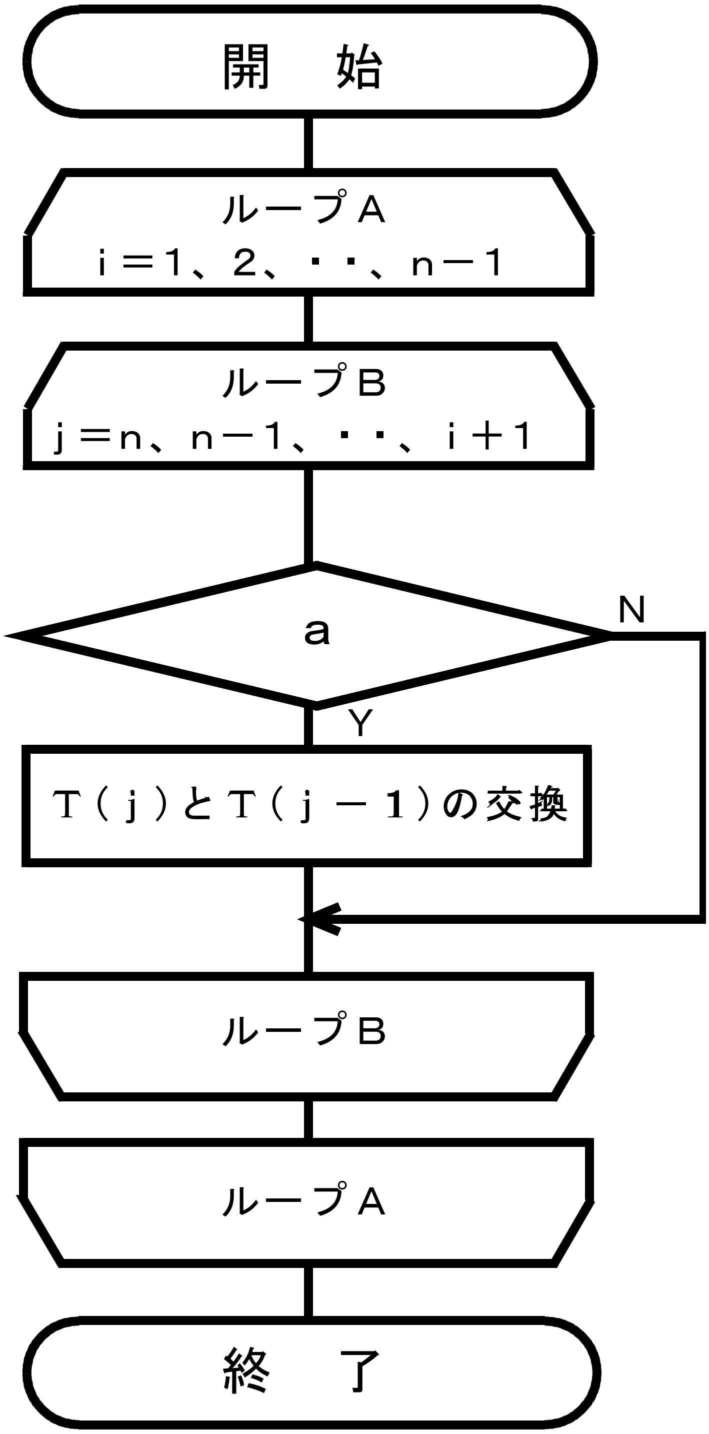
　ヒープソートは、親が子より大きいか、または小さいかのいずれかの特徴を持った2分木を利用して整列する方法である。

　アはシェルソートの説明でありクイックソートではない。
　イはバブルソートの説明でシェルソートではない。
　ウはクイックソートの説明であり、バブルソートの説明でない。
　エはヒープソートの説明であり、正しい。求める答えはエとなる。

◇問18解答　エ

問19問題

整数値からなるｎ個（ただし、ｎ≧２）のデータが、配列Ｔに格納されている。次の流れ図は、それらのデータを交換法を用いて昇順に整列する処理を示す。流れ図中のａに入れる適切な条件はどれか。



ア　Ｔ（ｊ）　＜　Ｔ（ｊ＋１）
イ　Ｔ（ｊ）　＜　Ｔ（ｊ－１）
ウ　Ｔ（ｊ）　＞　Ｔ（ｊ＋１）
エ　Ｔ（ｊ）　＞　Ｔ（Ｊ＋１）

◇問19解説

基本交換法の流れ図に関する問題である。

昇順に整列する基本交換法の手順

① 隣り合った数値を比較して大小関係が異なっていると交換する。
② 全データの一通りの大小の比較で最も大きいデータの整列位置が決まる。
③ 残りのデータについて同様の比較を繰り返す。
④ 全比較回数はデータ数の２乗に比例する。
⑤ 流れ図は二重ループになる。

ループＢが１回の一通りのデータの比較に相当する。流れ図では、データの比較は後方から前方に向かって行われており（ｊ＝ｎ，ｎ－１，…，２，１）、後のもの（Ｔ（ｊ））と一つ前のもの（Ｔ（ｊ－１））を比較し、Ｔ（ｊ）＜Ｔ（ｊ－１）ならば交換し、そうでなければ何もしない方式になっている。
従って、ａの内容はＴ（ｊ）＜Ｔ（ｊ－１）で、求める答えはイとなる。

◇問19解答　イ

問20問題

図はある配列をソートしたときに要素の順序が変わっていく様子である。

このソートのアルゴリズムは、次のうちのどれか。

（初期状態）	４、３、７、６、２、１、５
	$\boxed{3}$、４、７、６、２、１、５
	３、４、$\boxed{6}$、７、２、１、５
	$\boxed{2}$、３、４、６、７、１、５
	$\boxed{1}$、２、３、４、６，７、５
（整列後）	１、２、３、４、$\boxed{5}$、６、７

ア　クィックソート
イ　選択ソート
ウ　挿入ソート
エ　バブルソート

◇問20解説

挿入ソートの操作手順に関する問題である。

初期状態で整列済みのデータは４のみである。

１回目に３が対象になって４の前に設定される。

２回目は整列済みの３，４に７を挿入する。この場合は交換する必要がない。

３回目は整列済みの３，４，７に６を挿入する場合で、６は４と７の間に挿入される。

４回目は２を挿入する場合で、７と２、６と２、４と２、３と２を順次交換し、整列済みの先頭に挿入される。

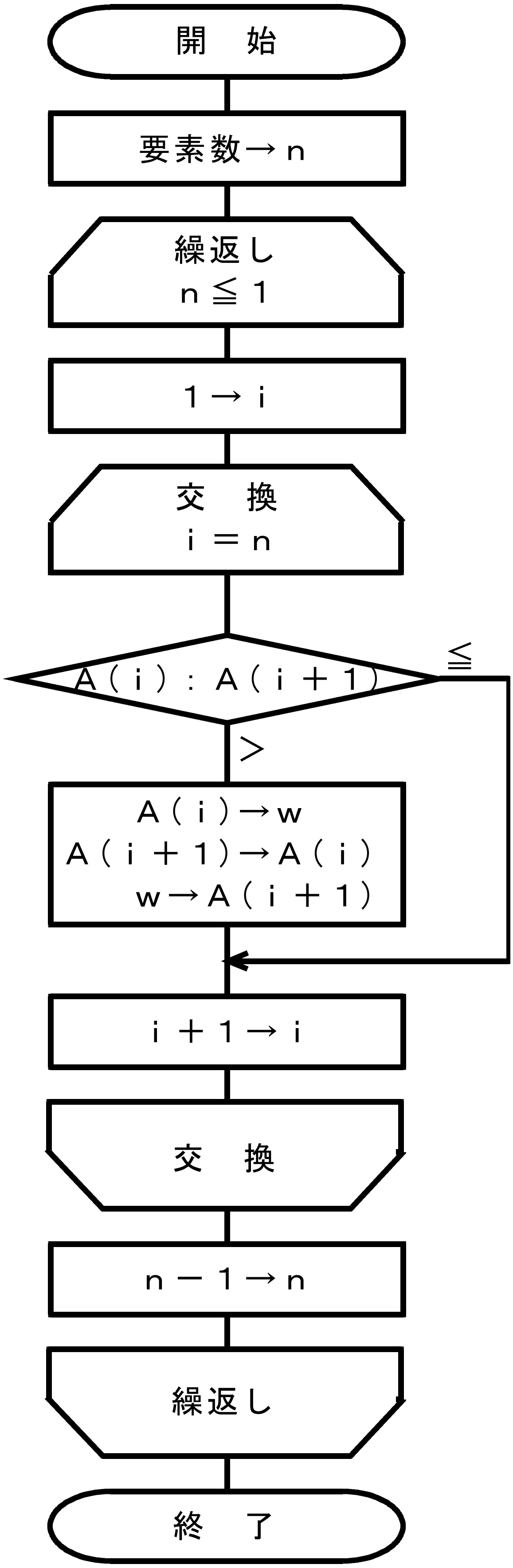
５回目は１を挿入する場合で、７と１、６と１、４と１、３と１、２と１を順次交換し、整列済みの先頭に挿入される。

６回目は５を挿入する場合で４と５の間に挿入される。

以上の操作は挿入ソートで、求める答えはウとなる。

◇問20解答　ウ

問21問題



流れ図が表す整列アルゴリズムはどれか。

ア　クィックソート
イ　シェルソート
ウ　挿入ソート
エ　バブルソート

◇問21解説

流れ図から、整列アルゴリズムを判定する問題である。

この流れ図は、ｉ＝１から隣り合う配列の大小を比較して、Ａ（ｉ）＞Ａ（ｉ＋１）ならば交換し、全要素の数ｎを比較して最大の要素一つを決定する。再び、残りのｎ－１個に対して同じ操作を繰り返して、ｎ≦１になるまでｎ－１回の操作を繰り返す。

この方法によって配列を昇順に整列するアルゴリズムで、バブルソートである。求める答えはエとなる。

◇問21解答　エ

問22問題

次の手順はシェルソートによる整列を示している。

データ列“７，２，８，３，１，９，４，５，６”を手順（１）～（４）に従って整列すると，手順（３）を何回繰り返して完了するか。ここで，［　］は小数点以下を切り捨てる。

〔手順〕

（１）　［データ数÷３］→Ｈとする。
（２）　データ列を互いにＨ要素分だけ離れた要素の集まりからなる部分列とし，それぞれの部分列を挿入法を用いて整列する。
（３）　［Ｈ÷３］→Ｈとする。
（４）　Ｈが０であればデータ列の整列は完了し，０でなければ（２）に戻る。

ア　２
イ　３
ウ　４
エ　５

◇問22解説

シェルソートに関する問題である。

シェルソートは、挿入ソートを改良したもので、データ列の中からＸ間隔ずつ離れたデータを取り出して、その部分列のデータを挿入ソートの考え方を利用して整列する。

このＸ間隔のギャップを、始めは大きく、次第に小さくして最後に１にする。ギャップの決め方はいろいろあるが、よく用いられる方法にデータ数の１／２、１／４　、…、１とする方式を用

いる。

データ数が９であるから、９÷３→Ｈ＝３、従って、３だけ離れた要素を挿入法で整列する。

３　１　６　４　２　８　７　５　９

３÷３→Ｈ＝１となり、

１　２　３　４　５　６　７　８　９

１÷３→Ｈ＝０

手順３の実行回数は２回となる。求める答えはアとなる。

◇問22解答　ア

問23問題

６個の数値１８０，３１５，２８２，４１０，６４５，５２５を並べ替える。手順１～４は途中までの手順を示したものである。

手順４まで終わったときの結果はどれか。

手順１　並びの左側から順に，数値の１の位の値によって０～９のグループに分ける。

手順２　次に０のグループの数値を左側から順に取り出して並べ，その右側に１のグループ，以下順に２～９のグループの数値を並べていく。

手順３　手順２で得られた数値の並びの左側から順に，数値の１０の位によって０～９のグループに分ける。

手順４　手順２と同様に，０のグループの数値から順に並べる。

ここで，グループ内では，処理が行われた数値を左側から順に並べるものとする。

ア　180、282、315、410、525、645
イ　315、410、525、180、282、645
ウ　410、315、525、645、180、282
エ　645、525、410、315、282、180

◇問23解説

整列に関する問題である。

①　１の位の値によって、０～９のグループに分け、左から順に並べる。

180、410、282、315、645、525

② 数値の１０の位の０～９のグループに分け、左から順に並べる。

410、315、525、645、180、282

処理結果は②の並びになる。求める答えはウとなる。

◇問23解答 ウ

問24問題

　データ全体をある値より大きいデータと小さいか等しいデータに二分する。次の二分されたそれぞれのデータの集まりにこの操作を適用する。これを繰り返してデータ全体を大きさの順に並べ替える整列法はどれか。

ア　クィックソート
イ　バブルソート
ウ　ヒープソート
エ　マージソート

◇問24解説

　クイックソートに関する問題である。

　アのクイックソートは、特定のデータＮ以下のデータ系列とＮ以上のデータ系列に分割する。データＮを基準にして、その軸の値よりも小さい要素は軸より前方へ、大きな要素は軸より後方へ振り分ける。一つの系列が適当な大きさになるまでこれを繰り返し整列する。求める答えはアとなる。
　イのバブルソートは、配列の隣り合ったデータを比較して、順序が違っていると並べ換えていく方法である。
　ウのヒープソートは、ソートすべきデータをヒープ構造の木を利用して並べ換える。
　エのマージソートは、整列済みの２つの系列を合わせて、一つの系列をつくるような方法で整列する。

◇問24解答　ア

問25問題

クイックソートの処理方法を説明したものはどれか。

ア　既に整列済みのデータ列の正しい位置に，データを追加する操作を繰り返していく方法である。
イ　データ中の最小値を求め，次にそれを除いた部分の中から最小値を求める。この操作を繰り返していく方法である。
ウ　適当な基準値を選び，それより小さな値のグループと大きな値のグループにデータを分割する。同様にして，グループの中で基準値を選び，それぞれのグループを分割する。この操作を繰り返していく方法である。
エ　隣り合ったデータの比較と入替えを繰り返すことによって，小さな値のデータを次第に端の方に移していく方法である。

◇問25解説

整列アルゴリズムのクィックソートに関する問題である。

クイックソートは、軸となるデータに着目し、そのデータより大きい系列と小さい系列に分類する。新しくできた二つの系列に対してそれぞれ軸となるデータを設定し、大きい系列と小さい系列に分額する。この操作を繰り返しながら対象のデータを昇順または降順に並べ替える方法である。操作の展開には再帰の考え方が用いられる。

アは基本挿入法、イは基本選択法、ウはクィックソート、エは基本交換法である。求める答えはウとなる。

◇問25解答　ウ

問26問題

　クィックソートに必要な考え方で、手続き中、自分自身を呼び出すのはどれか。

ア　リューザブル
イ　リエントラント
ウ　リロケータブル
エ　リカーシブ

◇問26解説

　クィックソートとリカーシブの関係に関する問題である。

　リカーシブは、プログラムの中から自分自身を呼び出すことを言う。自分自身を定義するのに自分自身よりも１次低い集合を用いる。その部分集合はより低次の部分定義を用いて定義することを繰り返して表現する。

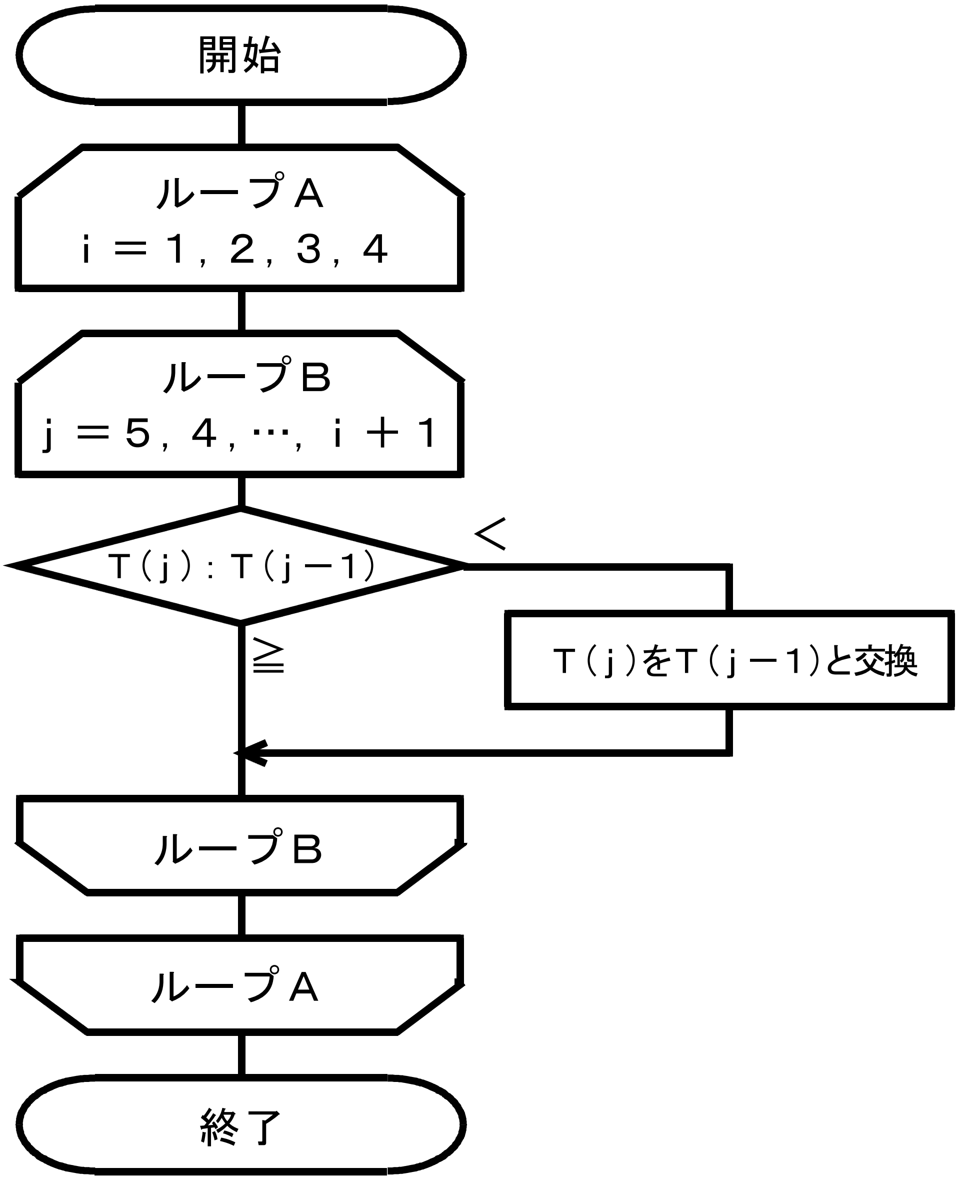
　クィックソートは、適当にサンプリングして得られたデータＮに着目し、Ｎ以下のデータ系列とＮ以上のデータ系列に分割する。このデータＮを軸(ピボット)という。データＮを軸として、その軸の値より小さい要素は軸より前方へ、大きな要素は軸より後方へ振り分ける。適当な長さの系列になるまで分割を繰り返し、それぞれの系列をソートすれば１つのソート済み系列となる。

　分割の繰返しに再帰呼び出し(リカーシブ)を利用した高速の整列法である。
　求める答えはエとなる。

◇問26解答　エ

問27問題

　次の流れ図で示されるアルゴリズムで配列Ｔを整列する。ｉ＝１でループＢが終わったときの配列Ｔの内容として、正しいものはどれか。



配列Ｔの初期値

Ｔ（１）	３１
Ｔ（２）	２
Ｔ（３）	２４
Ｔ（４）	１５
Ｔ（５）	４０

ア	イ	ウ	エ
２	２	３１	４０
２４	３１	２４	３１
１５	１５	１５	２
３１	２４	４０	２４
４０	４０	２	１５

◇問27解説

バブルソートの流れ図に関する問題である。

ソートする配列

３１　２　２４　１５　４０

に対して規則を用いると次のようになる。

３１　２　２４　１５　４０
比較
３１　２　２４　１５　４０
比較・交換
３１　２　１５　２４　４０
比較
３１　２　１５　２４　４０
比較・交換
２　３１　１５　２４　４０

ｉ＝１の操作が終了する。
求める答えはイとなる。

◇問27解答　イ

問28問題

　マスタファイルとトランザクションファイルを照合して、トランザクションファイルの情報でマスタファイルの変動項目の更新を行う処理はどれか。

ア　マージ
イ　マッチング
ウ　アップデート
エ　メンテナンス

◇問28解説

　複数ファイル処理の更新に関する問題である。

　アのマージは、同じファイル形式の複数のファイルを一つに統合することである。
　イのマッチングは、複数のファイルを突き合わせて２つ以上のデータが等しいかどうか検査することである。
　ウのアップデートは、ファイル、プログラム、データベースなどに修正、追加、削除を行うことである。
　エのメンテナンスは、ファイルの追加、更新や再編成である。

　ここでは、トランザクションファイルを利用して、マスタファイルの変動項目を更新する処理であるからアップデートである。求める答えはウとなる。

◇問28解答　ウ

問29問題

配列Ａ［ｉ］（ｉ＝１，２，…，ｎ）を，次のアルゴリズムによって整列する。行２～３の処理が初めて終了したとき，必ず実現されている配列の状態はどれか。

〔アルゴリズム〕

行番号
１　ｉを１からｎ－１まで１ずつ増やしながら行２～３を繰り返す

２　ｊをｎからｉ＋１まで１ずつ減らしながら行３を繰り返す

３　もしＡ［ｊ］＜Ａ［ｊ－１］ならば，Ａ［ｊ］とＡ［ｊ－１］を交換する

ア　Ａ［１］が最小値になる。
イ　Ａ［１］が最大値になる。
ウ　Ａ［ｎ］が最小値になる。
エ　Ａ［ｎ］が最大値になる。

◇問29解説

整列のアルゴリズに関する問題である。

後方からバブルソートで昇順に整列するアルゴリズである。

この場合の比較は、基準値Ａ［ｊ］と一つ前の要素Ａ［ｊ－１］と比較して、基準値が小さい場合に交換する。前の要素が大きく、後ろの要素が小さい場合に交換する。前に小さな値がくる整列である。

例えば、

２、４、１、５、３

の数列について考える。

ｉ＝１の場合、

　　２、４、１、<u>５、３</u>
→　２、<u>４、１</u>、３、５
→　<u>２、１</u>、４、３、５
→　１、２、４、３、５

となる。

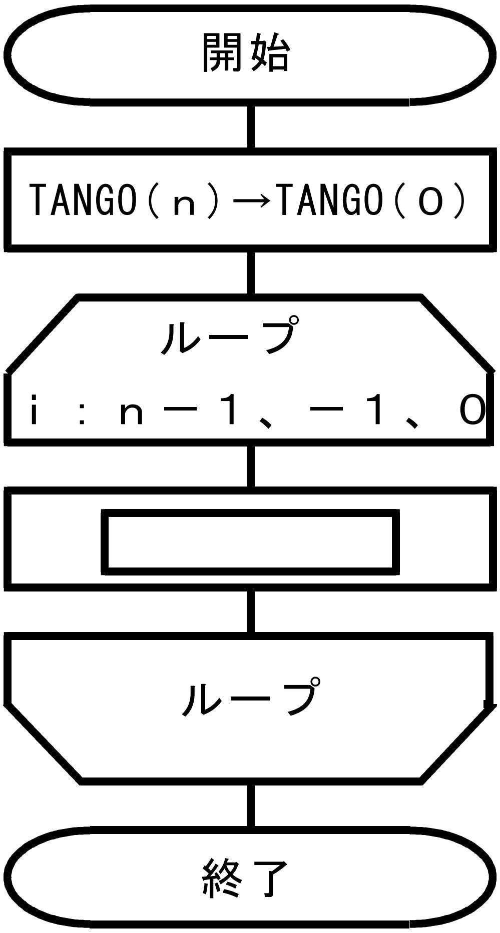
ｉ＝２の場合、

２、４、３、５の数列をｉ＝１の場合と同様にして、５から３までを比較して交換し、最小値２を求める。

以下、ｉ＝４まで、同様の操作を繰り返して昇順に整列する。従って、Ａ［１］が最小値になる。求める答えはアとなる。

◇問29解答　ア

問30問題



（注）ループにおける条件は、

変数名：初期値、増分、終値を示す。

要素番号が０から始まる配列TANGOがある。ｎ個の単語がTANGO(１)からTANGO(ｎ)に入っている。図は、ｎ番目の単語をTANGO(１)に入れるために、TANGO(１)からTANGO(ｎ－１)の単語を順に一つずつ後ろにずらして単語表を再構成する流れ図である。
　　　　に入れる処理として正しいものはどれか。

ア　TANGO(ｉ)→TANGO(ｉ＋１)
イ　TANGO(ｉ)→TANGO(ｎ－１)
ウ　TANGO(ｉ＋１)→TANGO(ｎ－１)
エ　TANGO(ｎ－ｉ)→TANGO(ｉ)

◇問30解説

配列の要素の移動に関する問題である。

同じ配列上で１文字後方に移動させる場合、配列の後方から移動させる。

TANGO(n)をTANGO(0)に格納し、次の手順でｎ－１～０までをｎ～１に移動させる。

大きさＮの配列ＴＢＬ（Ｉ）を１文字後方に移動する手順

① Ｉにｎ－１を格納する。
② TANGO（Ｉ）→TANGO（Ｉ＋１）を実行する。
③ Ｉ－１→Ｉを実行する。
④ Ｉ≧０ならば②に戻る。
⑤ Ｉ＜０になると、移動を終了する。

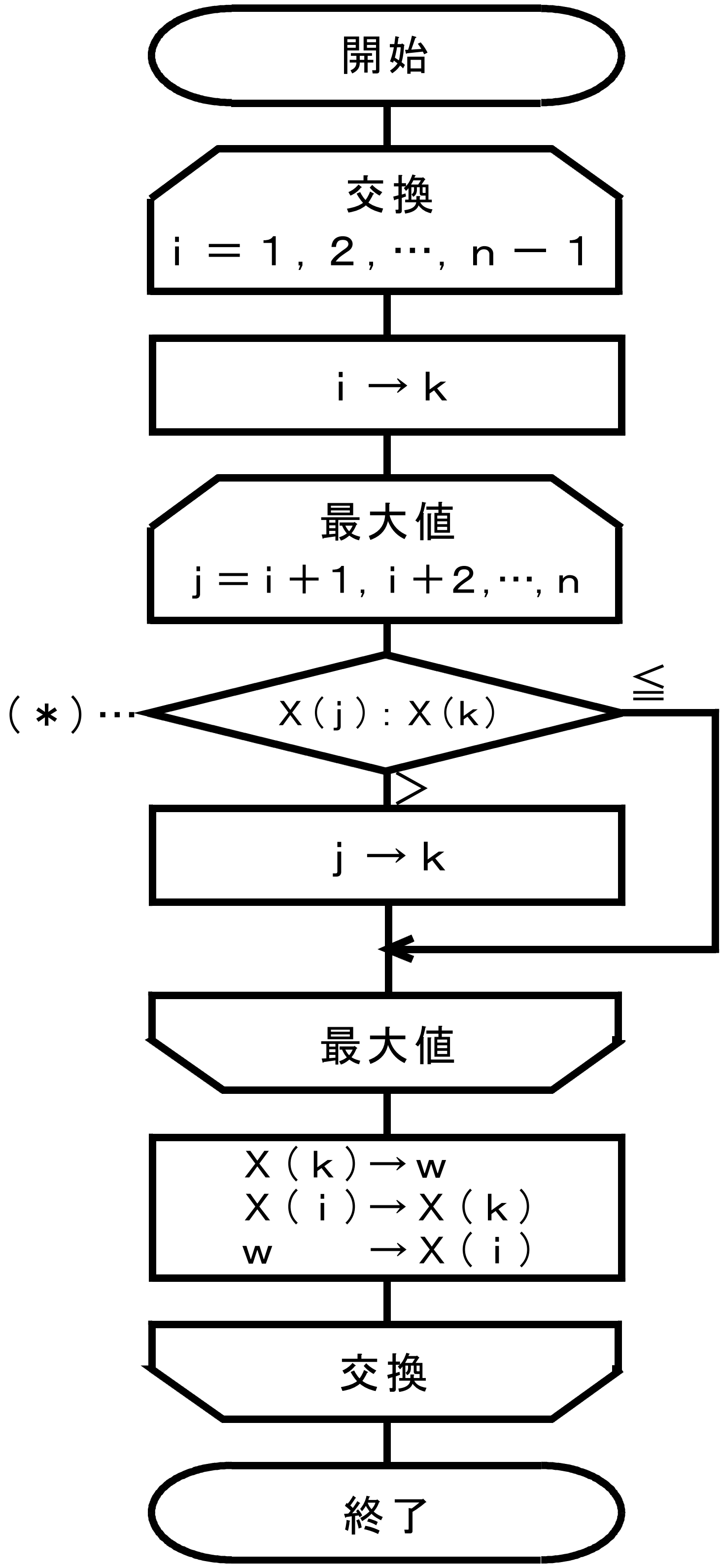
□の中に入れる処理は

TANGO（Ｉ）→TANGO（Ｉ＋１）

となり、求める答えはアとなる。

◇問30解答　ア

問31問題



流れ図は、最大値選択法によって値を大きい順に整列するものである。＊印の処理（比較）が実行される回数を表す式はどれか。

ア　ｎ－１
イ　ｎ（ｎ－１）／２
ウ　ｎ（ｎ＋１）／２
エ　n^2

◇問31解説

基本選択法に関する問題である。

基本選択法の手順（降順に整列する場合）

① 配列中の要素を左から順次比較することにより、配列内の最大値を求める。
② その最大値が格納されている要素と左端の要素を入れ替え、左端に最大値を求める。
③ 左端の位置を右に一つずらし、再度左端から順次比較し最大値を求める。
④ 以上の処理を要素数Ｎ－１回繰り返し、昇順に整列される。

流れ図の最大値の繰り返しの内容

１回の最大値を決めるための操作であり、この最大値の繰り返しが次に示すようにｎ－１回繰り返されることになる。

流れ図の交換の繰り返しの内容

１回目の最大値を求める場合の比較回数は、データ数をｎとすると、ｎ－１回となる。
２回目の最大値を求める場合の比較回数は、データ数がｎ－１で、ｎ－２回となる。
３回目の最大値を求める場合の比較回数は、データ数がｎ－２で、ｎ－３回となる。

同様にして

　ｎ－２回目の最大値を求める場合の比較回数は、データ数が３で、２回となる。

　ｎ－１回目の最大値を求める場合の比較回数は、データ数が２で、１回となる。

　以上のｎ－１回の最大値の繰り返しの比較回数の総数Ｓは次のようになる。

Ｓ＝（ｎ－１）＋（ｎ－２）＋（ｎ－３）＋…＋３＋２＋１

　次のようにして計算する。

Ｓ＝（ｎ－１）＋（ｎ－２）＋（ｎ－３）＋…＋３＋２＋１　……①

Ｓ＝１＋２＋３＋…＋（ｎ－３）＋（ｎ－２）＋（ｎ－１）　……②

①＋②を求めると

２Ｓ＝ｎ＋ｎ＋ｎ＋…＋ｎ＋ｎ＋ｎ

（ｎがｎ－１回加算される）

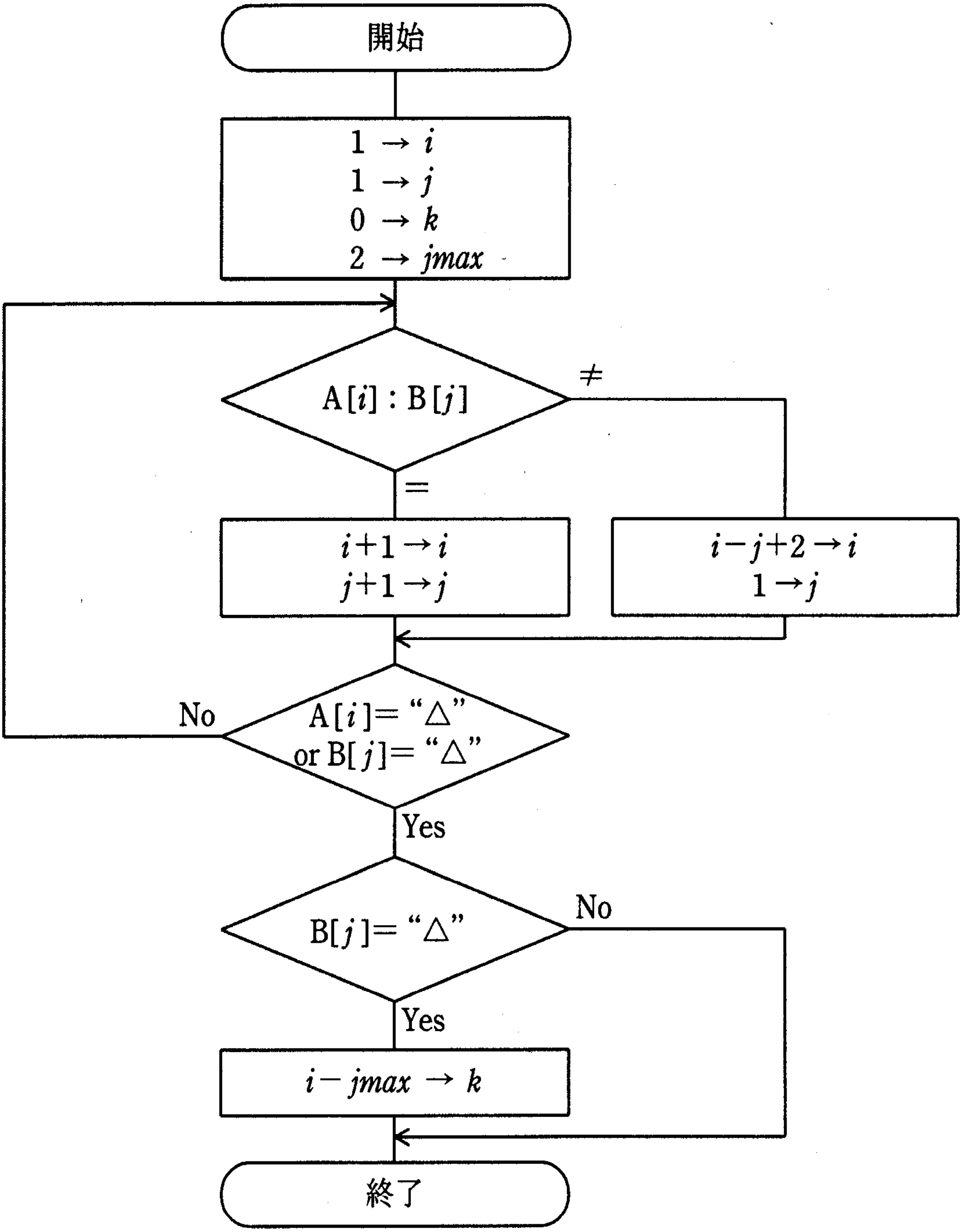
　　＝ｎ（ｎ－１）

　Ｓ＝ｎ（ｎ－１）／２

となり、求める答えはイとなる。

◇問31解答　イ

問32問題


開始
1 → i
1 → j
0 → k
2 → jmax
A[i] : B[j]
≠
=
i+1 → i
j+1 → j
i−j+2 → i
1 → j
No
A[i]= "△"
or B[j]= "△"
Yes
B[j]= "△"
No
Yes
i − jmax → k
終了

　文字列Ａが“ａａｂａｂｘ△”，文字列Ｂが“ａｂ△”であるとき，流れ図の終了時点のｋは幾らか。

　ここで，文字列の先頭の文字を１番目と数えるものとし，Ａ［ｉ］はＡのｉ番目の文字を，Ｂ［ｊ］はＢのｊ番目の文字を，“△”は終端を示す文字を表す。

ア　０
イ　１
ウ　２
エ　４

◇問32解説

　流れ図に関する問題である。

　配列に格納された２つの文字列を照合する問題である。

　照合の手順

　①　文字列Ａ、文字列Ｂの先頭から対応する要素を比較する。

　②　２つの要素が一致すると、文字列Ａ、文字列Ｂ共に添字のインクリメントを行い、④に移る。

　③　２つの要素が一致しない場合、ｉ－ｊ＋２→ｉで文字列Ａの先頭位置を求め、文字列Ｂは先頭位置に戻る。

　④　文字列Ａまたは文字列Ｂのいずれかがスペースになると照合を終了し、スペースならない場合、①の処理に戻る。

　⑤　照合が終了した場合、文字列Ｂのスペースで終了すると、ｉ－ｊｍａｘ→ｋを求める。そうでなければ何もしないで終

了する。

文字列Ａ＝ａａｂａｂｘ△、文字列Ｂ＝ａｂ△として照合手順を実行する。

① ａ＝ａで一致

② ａ≠ｂで不一致、ｉ＝２、ｊ＝２、ｉ－ｊ＋２→２となる。

③ ａ＝ａで一致

④ ｂ＝ｂで一致

⑤ ａ＝△で不一致、ｉ＝４、ｊ＝３、ｉ－ｊ＋２→３となる。

⑥ ｂ［３］＝△で照合を終了し、ｉ＝４、ｊｍａｘ＝２であるから、ｋ＝４－２＝２となる。

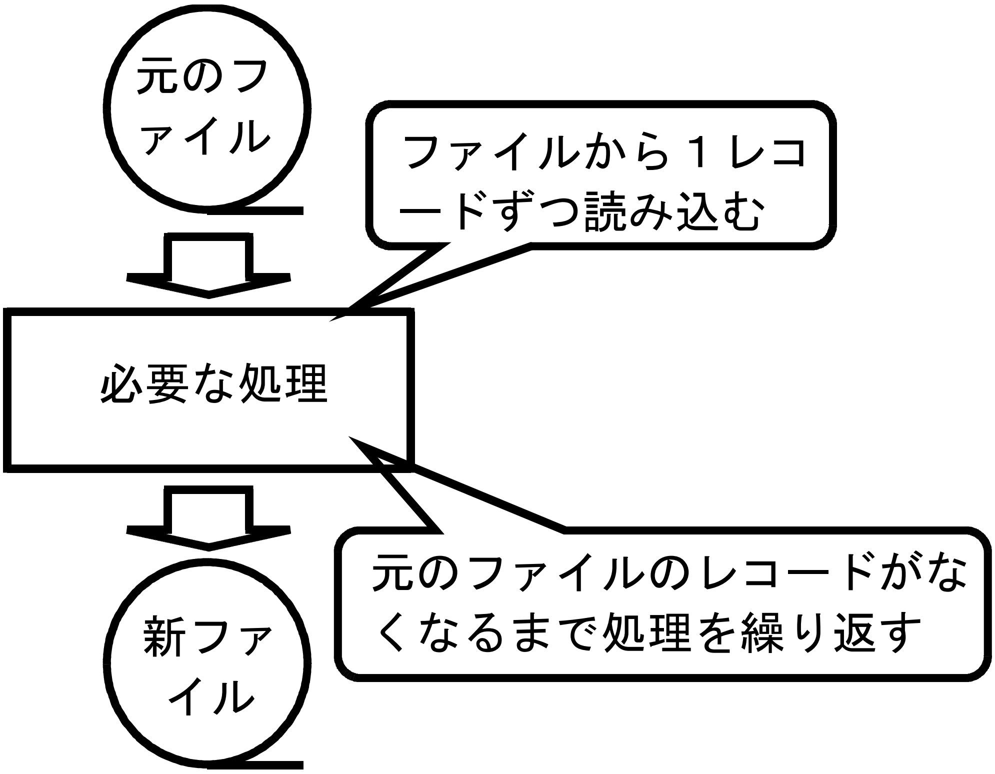
求める答えはウとなる。

◇問32解答　ウ

ファイル処理

◆単一ファイル処理

◇単一ファイル処理とは



単一ファイル処理は、複数のレコードが格納されているファイルから、１レコードずつ読込み、媒体変換やレコードの検査、グループ集計処理などを行つて新しいファイルを作成したり、一定の様式の帳票を出力する処理である。

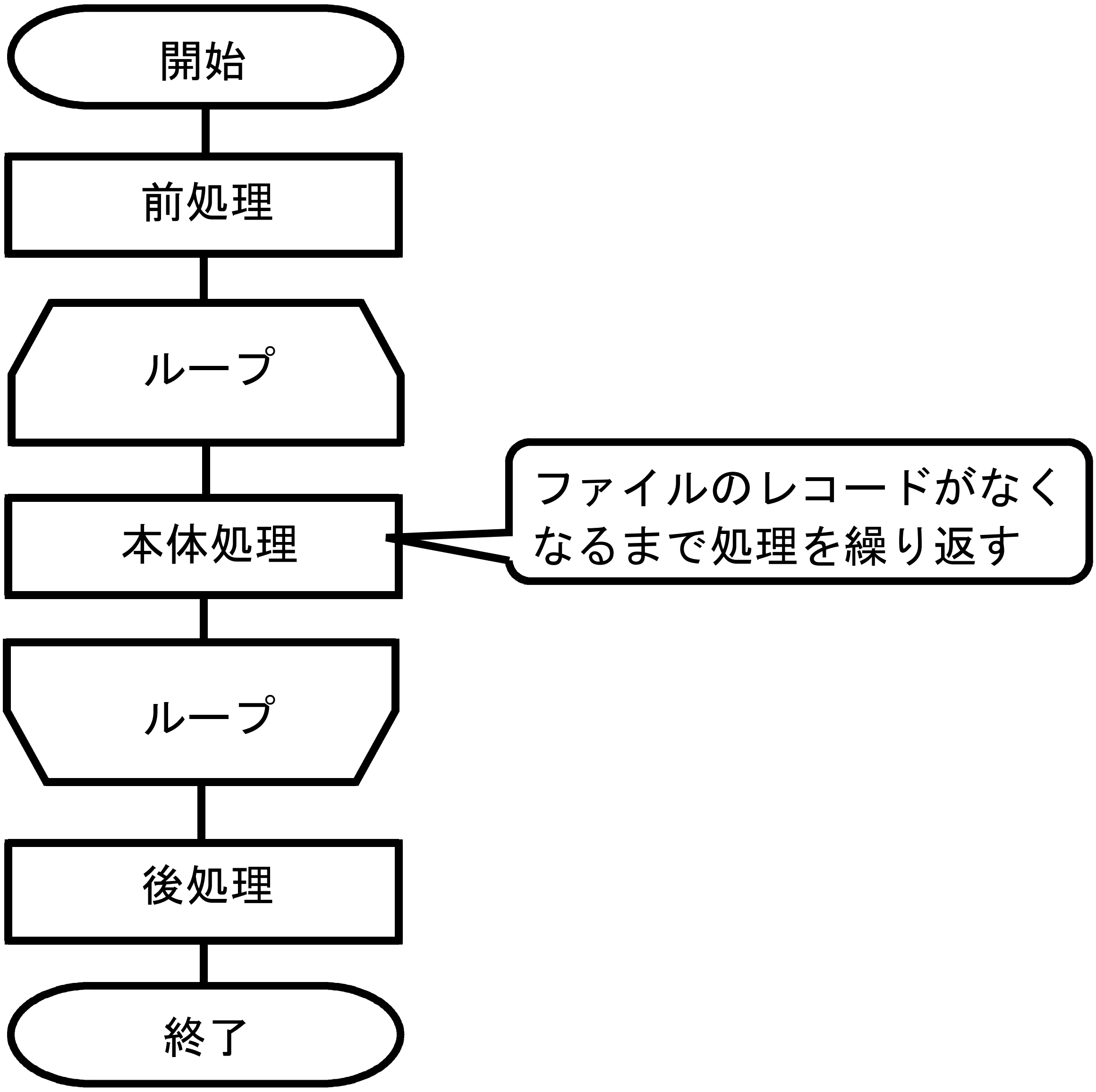
ファイル処理を実行する前に、ファイル内のレコードの整列を行う。グループ集計処理にはコントロールブレーク処理が利用される。

◇アルゴリズムの構成

単一ファイル処理のアルゴリズムは、次の３部分に分けることができる。

① 前処理
② 本体処理
③ 後処理

本体処理では、複数のレコードに対して同じ処理を繰り返す。



◇前処理で実行される内容

① 必要な記憶領域の確保

ⓐ 入力データの記憶領域
ⓑ 演算結果の記憶領域
ⓒ 出力データの記憶領域などの

② 記憶領域の初期化

③ 関連するファイルのオープン

④ 最初のデータの入力、最初の表題行、見出し行の印刷

⑤ キーの記憶域への移動

⑥ 最初のデータの演算処理、演算結果の記憶領域への出力

◇本体処理で実行される内容

① 入力レコードの有無の判断

② 仕事の処理内容の分類

③ キーの比較、キーの記憶域への移動

④ 必要な演算処理、演算結果の記憶領域への出力、報告書やファイルへの出力

⑤ データ項目の照合、更新、削除、追加処理

⑥ レコードの整列、併合処理

⑦ 改ページ制御、空白行の制御、表題・見出し行の印刷

⑧　処理件数、処理行のカウント

⑨　データの初期化、次のデータの入力

◇本体処理の特徴

①　一定の処理をレコードがなくなるまで繰り返す。

②　幾つかの機能を分類したり、組み合わせたりしながら、繰り返し順次処理を進める。

③　繰り返し処理が中子になったり、条件に従って幾つかの処理が選択的に行われる。

④　グループ集計では特定のキーが利用される。

◇後処理で実行される内容

①　必要な演算処理

②　処理結果の出力

③　関連ファイルのクローズ

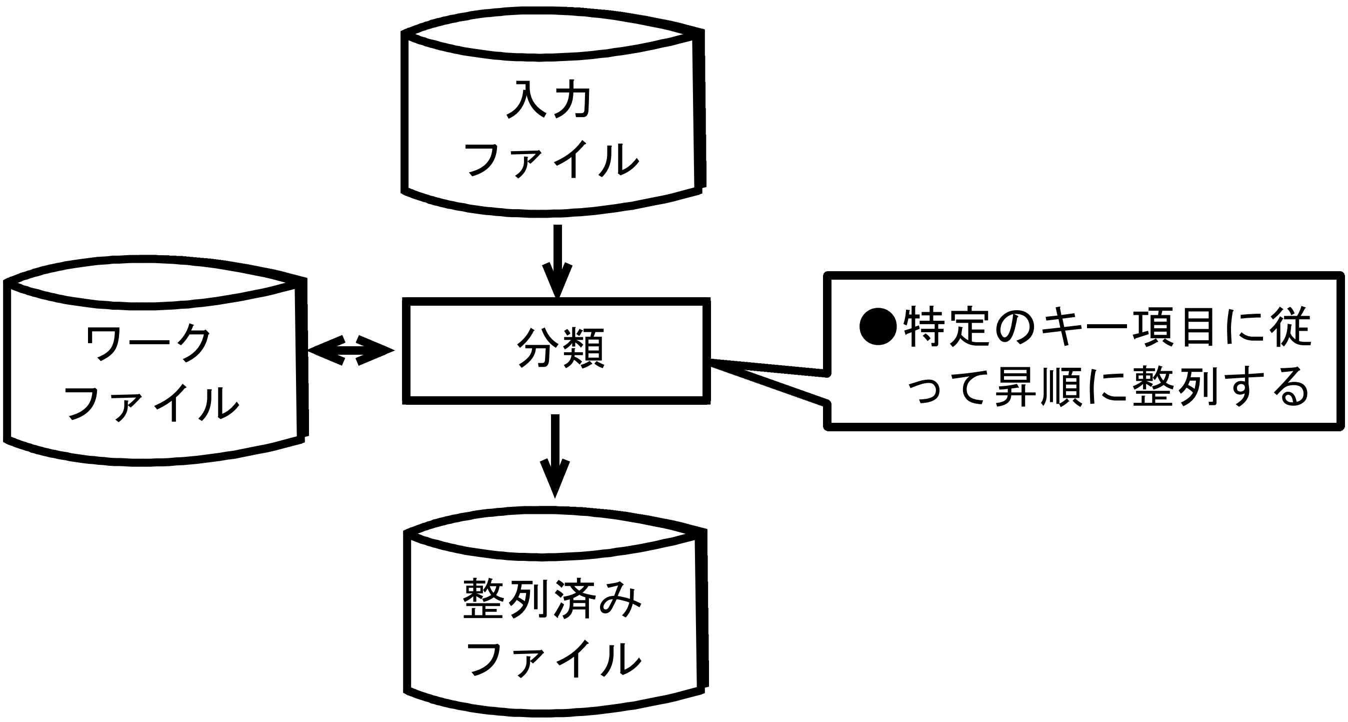
◆単一ファイル処理の基本パターン

◇入力媒体変換

入力媒体変換は、処理プロセスに最適な媒体に変換する処理で、入力媒体上のファイルを他の記録媒体に記録し直す。

◇分類処理（整列）

分類処理は、ファイルの中のレコードの順番を特定の項目の値や文字列の順番に従って並び換える処理である。特定の項目を整列キーという。



整列キーの小さいレコードから順に並び換えることを昇順に整列といい、その逆を降順に整列という。第一次キー、第二次キー、…、のように複数のキーを利用して整列することもある。処理以前の整列結果を利用して、最小限の整列キーを指定するだけで整列の目的が達せられることもある。

◇出力媒体変換

出力媒体変換は、記録媒体上のマスタファイルから、利用者向けの媒体に見やすい体裁に出力する。

対象になるファイルのレコードを１件ずつ印刷出力する明細行印刷などがある。

◇入力レコード検査

入力レコード検査は、プログラムによるデータのチェック法が使用される。

入力レコードを検査する方法には、数字チェック、形式チェック、限界チェック、論理チェック等の各種チェックの方法がある。

◇生成処理

１つ以上のファイルを読み込んで、レコードの変形や加工を行う。

非正規形のレコードを正規化したレコードに変換する処理などがある。

◇複写処理

入力ファイルのレコードを出力ファイルに書き写す処理を行う。

◇抽出処理

必要な情報を検索して抜き出す処理を行う。

◆グループ制御処理

◇グループ制御

グループ集計では、整列済みの入力レコードのキー項目を順次比較し、レコードのグループを見分け、グループ内の集計を行う。

整列対象のキー項目をコントロール項目、同じキー項目を持つレコード群をコントロールグループ、グループの変わり目をコントロールブレークという。

制御対象が多重の階層をなしている場合を多重グループ制御という。多重グループ制御では複数のキーが利用され、それらのキーが整列の場合、第１次キー、第２次キー、第３次キー、…、となる。第１次キーは第２次キーよりも上位のレベルにあり、第２次キーは第３次キーよりも上位のレベルにあり、以下同様なレベル関係にある。

多重レベルのコントロールブレーク処理を行う場合、上位のレベルのキーの変化から判定し、上位のキーが不変の場合、下位のキーの判定に移る。上位のレベルのキーが変化すると、下位のレベルのキーは自動的に変化する。

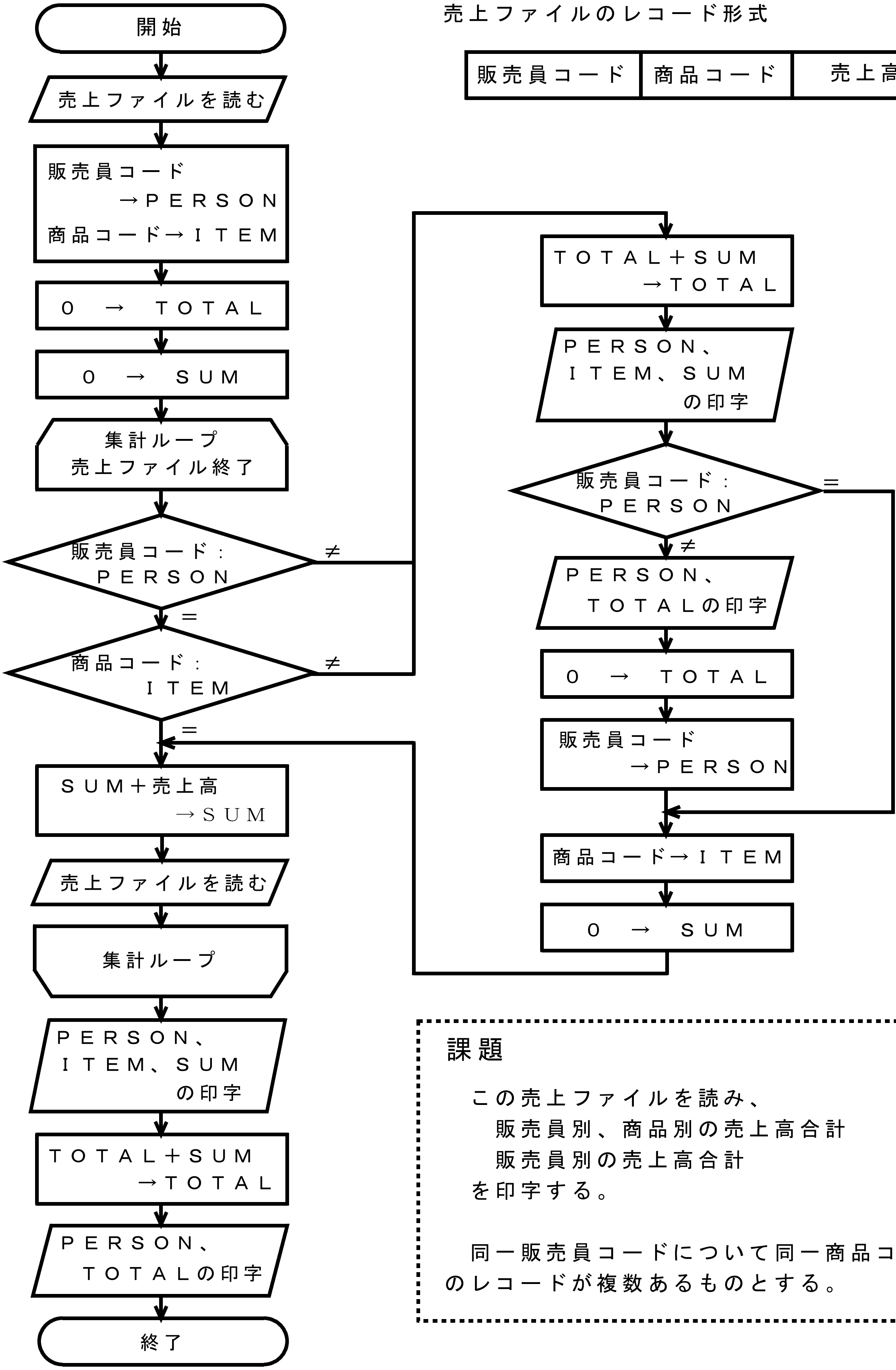
コントロールブレーク処理は、終わりの処理は下位のレベルから上位のレベルの順に処理し、初めの処理は上位のレベルから下位のレベルの順に処理を進める。

◇グループ集計処理の具体例

流れ図は、販売コード、商品コード、売上高からなる売上げファイルのレコードを読み込み、販売員別商品別売上高合計、販売員別売上高合計を印字するプロセスフロー図である。

売上ファイルのレコード形式

販売員コード	商品コード	売上高



課題

この売上ファイルを読み、
　販売員別、商品別の売上高合計
　販売員別の売上高合計
を印字する。

同一販売員コードについて同一商品コードのレコードが複数あるものとする。

第１次キーが販売員コード、第２次キーが商品コードでグループ集計処理が行われる。

ＰＥＲＳＯＮ、ＩＴＥＭは、コントロールブレーク処理の比較基準になる販売員コード、商品コードを格納する記憶領域である。

コントロールブレーク処理は、第１レベルの販売員コードの変化を判定し、販売員コードが不変の場合に、第２レベルの商品コードの変化を判定する多重レベルのコントロールブレーク処理になっている。

売上高実績表

単位　万円

販売員コード	商品コード	売上高	売上高小計
１００１	１０１０	１２００	
	１０２０	３１００	４３００
１００２	１０１０	１５００	
	１０２０	２７００	４２００
		売上高合計	８５００

流れ図の集計ループでは、レコードの販売員コードとＰＥＲＳＯＮを比較し、同一ならば、商品コードとＩＴＥＭを比較し、同一の間、集計を繰り返す。

途中で、商品コードが変化すると、新しい商品コードの集計を行う。この間、同一販売員コードに関して、複数の商品コードの集計が実行される。

第一キーの販売員コードが変化すると、最初の商品コードに戻り、商品別売上高の集計が行われる。

結果として、売上高実績表に示す販売員コード別、商品コード別売上高が集計されることになる。

◆複数ファイル処理

◇複数ファイル処理とは

　複数ファイル処理は、入力ファイルが二つ以上の時、各ファイルのレコード間の論理関係を確認しながら処理を進める方法である。

　次の処理がある。

　①　突合せ
　②　併合
　③　更新
　④　維持

◇突合せ（マッチング）

　突合せは、基本となるファイルのレコードの中の特定のデータ項目を参照したり、取り出し、別のファイルのレコードの内容と合体させて、新しいレコードとする処理である。

◇併合（マージ）

　併合は、キー項目の大小関係に基づいて、複数個のファイルの中のレコードを昇順、または降順に組み合わせて、一つのファイルにまとめる処理である。

◇更新（アップデート）

　更新は、マスタファイルのレコードの変動項目を、トランザクションファイルのレコードの項目で更新する処理である。

◇維持（メンテナンス）

維持は、マスタファイルに新規レコードを追加したり、不要になったレコードを削除したり、レコードの固定的な項目の内容を変更したり、訂正したりして、マスタファイルを最新状態に維持する処理である。

◇マスタファイル

マスタファイルは、内容があまり変わらない基本になるファイルである。

社員マスタファイル、顧客マスタファイル、商品マスタファイルなどがある。

◇トランザクションファイル

トランザクションファイルは、内容が取引やレコードの発生のつど変化するファイルである。

残業時間ファイル、売上ファイルなどがある。

◆複数ファイル処理の基本

◇２つのファイルの整列条件

複数ファイル処理の対象となるファイルは、両ファイルとも特定の同一項目で整列していなければならない。

特定のデータ項目が複数存在する場合、第１キー、第２キー、…、が一致していなければならない。

◇２つのファイルのキー項目の比較

複数ファイル処理を実行する場合、ファイルＡのキー項目（Ａキー）とファイルＢのキー項目（Ｂキー）の大小関係を比較する。

①　Ａキー＝Ｂキーならば、

どちらかのファイルを利用して、他方のファイルの処理を実行する。

②　Ａキー＜Ｂキーならば、

ファイルＡに必要な処理を実行して、ファイルＡの次のレコードを読み込む。

③　Ａキー＞Ｂキーならば、

ファイルＢに必要な処理を実行して、ファイルＢの次のレコードを読み込む。

ＡキーとＢキーが一致しない場合は、基本的にはキー値の小さいファイルを対象に処理を行う。

処理の内容はファイル処理の種類、発生条件に応じて異なる。

◇ファイル処理の終了条件

２つのファイルの一方の処理が終了しても、他方のファイルの残りの処理を実行しなければならない。

１つのファイルの処理が終了すると、処理の終了したファイルの対象になっているキー項目に最大のキー値を設定する。

２つのファイルの対象のキー項目が共に最大値になったとき、２つのファイル処理を終了する。

ファイルのレコードがなくなると、キー項目に最大のキー値を設定するのは、両方のファイルにレコードがある場合と同じ処理手順で、そのまま実行できるためである。

レコードが両ファイルにある場合と、片側のファイルにしかない場合のアルゴリズムを同一にすることができる。

◆ファイルの併合

◇併合とは

特定のキー項目の大小関係に基づいて、複数個のファイル内のレコードを昇順または降順に組み合わせて、１つのファイルにまとめることである。

複数のファイルは同一キー項目で併合前に昇順に整列している。併合前後のファイルのレコード様式は共に同じ様式である。

◇２つのファイルを併合する手順

① Ａ、Ｂファイルから１レコードずつ読み込む。

② 読み込んだ２つのレコードのキー項目の大小関係を比較し、キー項目の小さい方のレコードをＣファイルのレコードとして出力する。

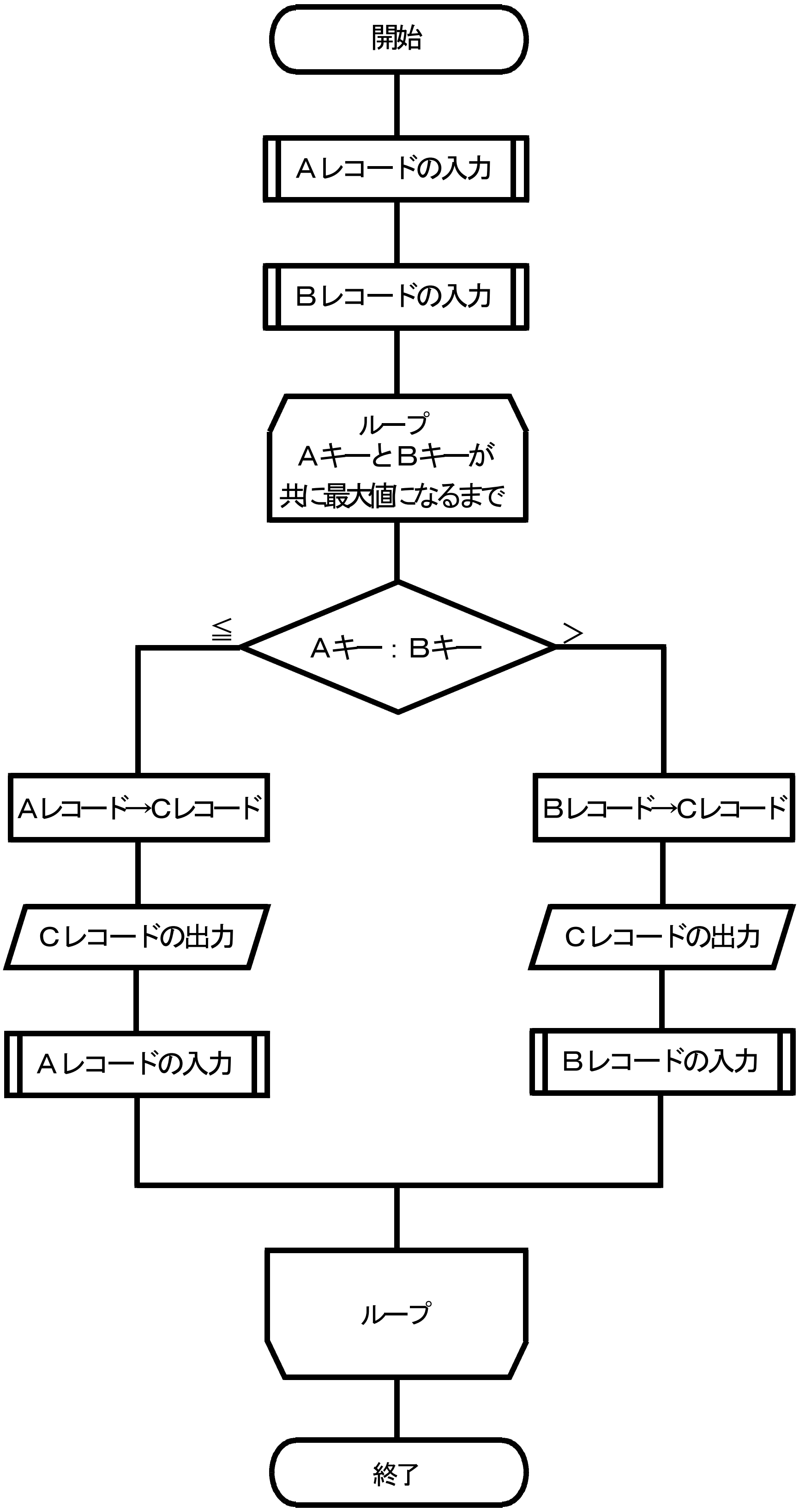
③ Ａファイル（Ｂファイル）から読み込まれたレコードをＣファイルに出力した場合は、次に読み込むレコードはＡファイル（Ｂファイル）から読込む。Ａファイル（Ｂファイル）から読み込んだレコードと前にＢファイル（Ａファイル）から読み込んだレコードとキーの値の大小関係を比較する。

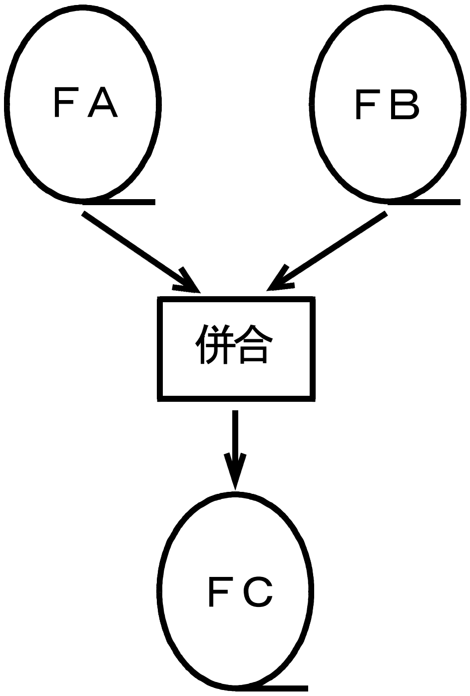
④ Ａ、Ｂファイルのキー項目の値が等しい場合は、あらかじめ決められた規則に従って処理を行う。

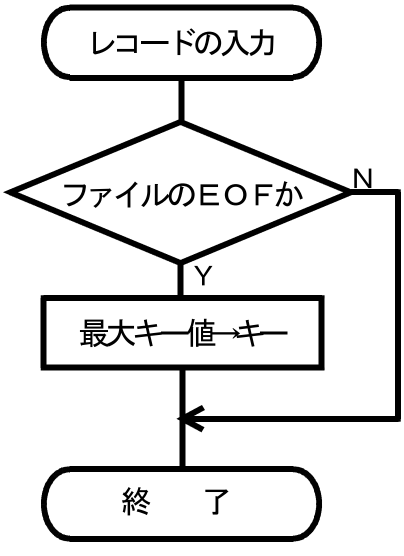
⑤ Ａ、Ｂのファイルのレコードがなくなるまで、キー項目の比較を繰り返す。

⑥ ファイルのレコードが終了すると、そのファイルのキー項目に最大値を設定する。

⑦　Ａ、Ｂのファイルが共に最大値になると、併合処理を終了する。







◆ファイルの突合せ

◇突合せとは

突合せは、ファイルＡとファイルＢの、特定のキー項目の一致するレコードの特定のデータ項目を照合や参照することである。

処理前に、ファイルＡ、ファイルＢ共に同一キー項目で昇順に整列している。

◇同一キーが複数個ある場合の手順

① Ａレコード、Ｂレコード共に同じ項目のＡキー、Ｂキーで昇順に整列する。

② ファイルＡ、ファイルＢからＡレコード、Ｂレコードをそれぞれ１レコード入力する。ファイルＡ、ファイルＢのレコードが無くなると、Ａキー、Ｂキーの値を最大値に設定する。

③ ＡキーとＢキーの値を比較する。

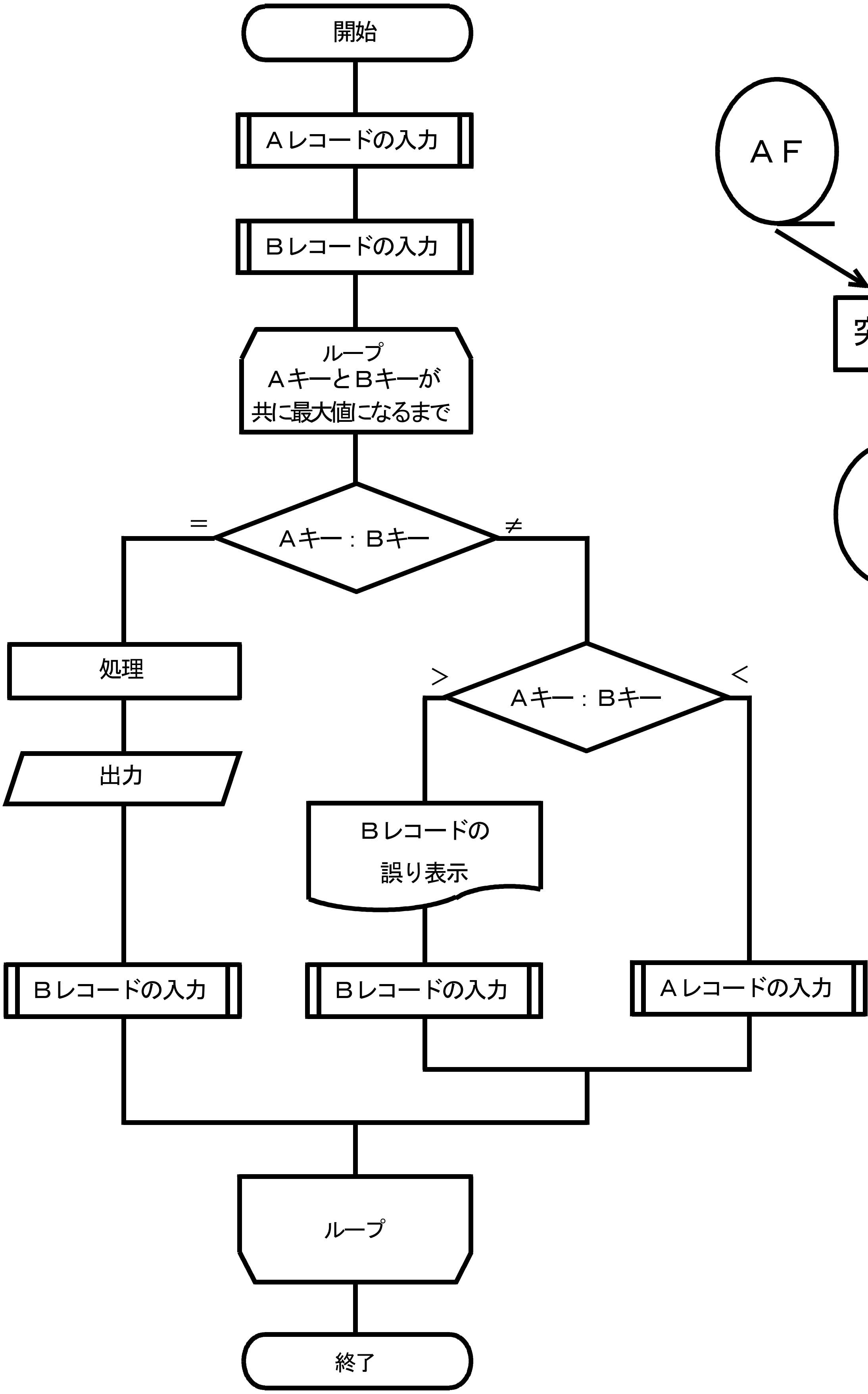
④ Ａキー＝Ｂキーならば、Ｂレコードに対応するＡレコードが存在するので、Ａレコードを利用してＢレコードに必要な処理を行い、そのＢレコードを出力し、次のＢレコードを入力する。

⑤ Ａキー＞Ｂキーならば、Ｂレコードに対応するＡレコードが存在しないため、そのＢレコードに対してエラー処理を行い、次のＢレコードを入力する。

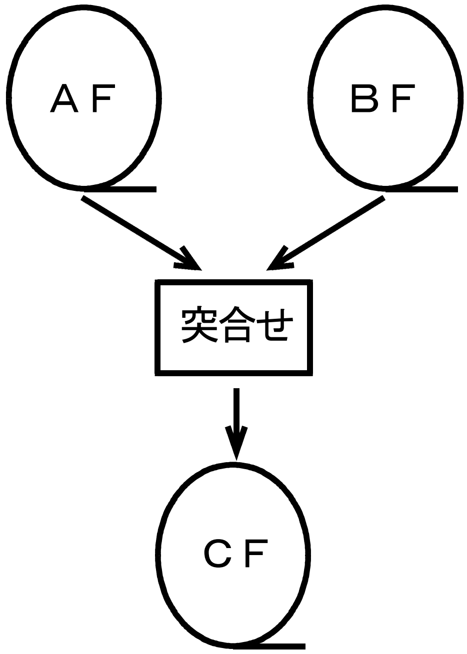
⑥ Ａキー＜Ｂキーならば、Ｂレコードに対応するＡレコードがそれ以降のファイルＡ内に存在する可能性があるため、Ｂレコードはそのままの状態で、次のＡレコードを入力する。

⑦ Ａキー、Ｂキーが共に最大値になると処理を終了し、そうで

なければ③に戻る。



開始
Ａレコードの入力
Ｂレコードの入力
ループ
ＡキーとＢキーが
共に最大値になるまで
Ａキー：Ｂキー
＝
≠
処理
出力
Ｂレコードの入力
Ａキー：Ｂキー
＞
＜
Ｂレコードの
誤り表示
Ｂレコードの入力
Ａレコードの入力
ループ
終了



ＡＦ
ＢＦ
突合せ
ＣＦ

◆ファイルの更新

◇更新とは

　ファイルＡとファイルＢの特定のＡキーとＢキーが一致すると、Ｂレコードの特定のデータ項目を利用して、Ａレコードの特定のデータ項目の更新処理を行う。

　処理前に、ファイルＡ、ファイルＢ共に同一キー項目で昇順に整列している。

◇同一キーが複数個ある場合の手順

①　Ａレコード、Ｂレコード共に同じ項目のＡキー、Ｂキーで昇順に整列する。

②　ファイルＡ、ファイルＢからＡレコード、Ｂレコードをそれぞれ１レコード入力する。ファイルＡ、ファイルＢのレコードが無くなると、Ａキー、Ｂキーの値を最大値に設定する。

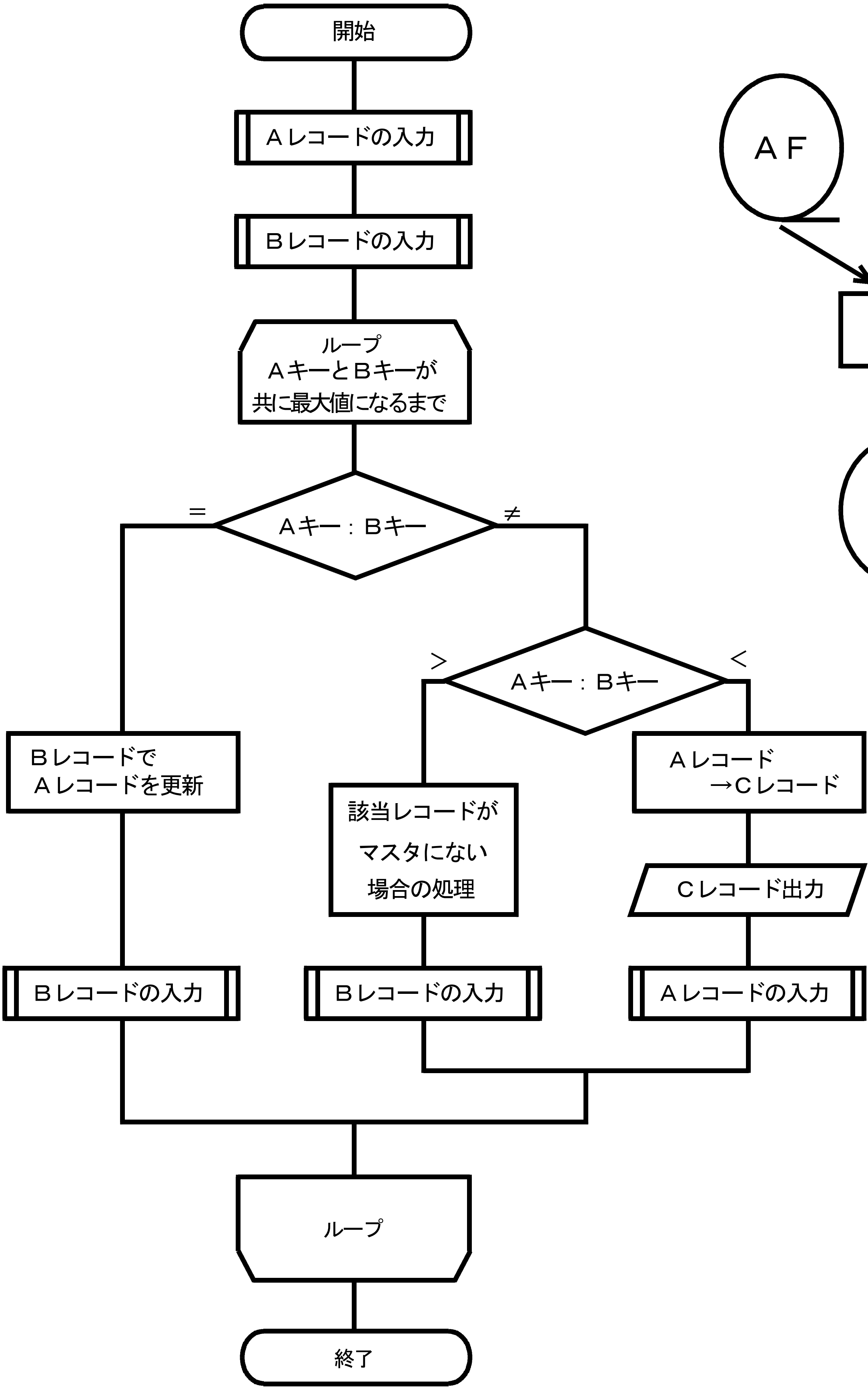
③　ＡキーとＢキーの値を比較する。

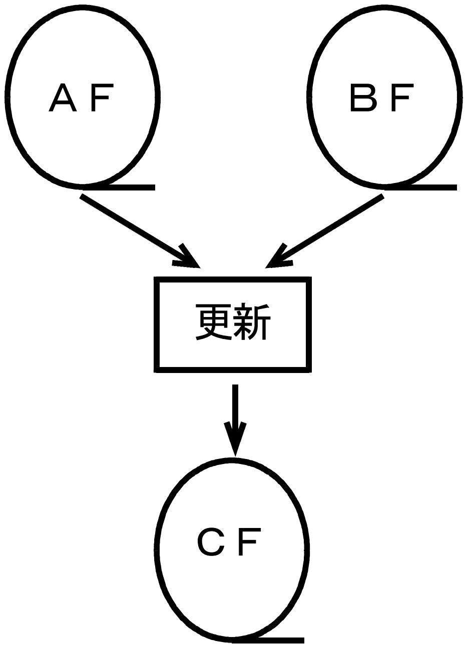
④　Ａキー＝Ｂキーならば、Ｂレコードの内容を利用してＡレコードの対応するデータ項目の更新処理を行い、Ａレコードはそのままの状態で、次のＢレコードを入力する。

⑤　Ａキー＞Ｂキーならば、Ｂレコードに対応するＡレコードが存在しないため、そのＢレコードに対してエラー処理を行い、次のＢレコードを入力する。

⑥　Ａキー＜Ｂキーならば、Ｂレコードに対応するＡレコードがそれ以降のファイルＡ内に存在する可能性があるため、Ｂレコードはそのままの状態で、Ａレコードを新しいファイルＣに出力し、次のＡレコードを入力する。

⑦　Ａキー、Ｂキーが共に最大値になると処理を終了し、そうでなければ③に戻る。





◆レコードの追加・削除

◇追加・削除

複数ファイル処理でファイルＢの内容を利用して、ファイルＡへのレコードの追加やファイルＡからのレコードの削除を行うことができる。

ファイルＡ、ファイルＢ共に同一キー項目で昇順に整列している。

◇同一キーがない場合の追加・削除

① Ａレコード、Ｂレコード共に同じ項目のＡキー、Ｂキーで昇順に整列する。

② ファイルＡ、ファイルＢからＡレコード、Ｂレコードをそれぞれ１レコード入力する。ファイルＡ、ファイルＢのレコードが無くなると、Ａキー、Ｂキーの値を最大値に設定する。

③ ＡキーとＢキーの値を比較する。

④ Ｂキー＝Ａキーならば、Ａキーのレコードをファイルから削除する。削除後、ファイルＢから新しいＢレコードを、ファイルＡから新しいＡレコードを入力する。

⑤ Ａキー＞Ｂキーならば、Ｂキーに対応するＡレコードがファイルＡ内に存在しないため、入力したＡレコードはそのままの状態で、Ｂレコードを利用して新しいＡレコードを作成し、ファイルＡに追加し、Ｂファイルから新しいＢレコードを入力する。

⑥ Ａキー＜Ｂキーならば、Ｂレコードはそのままの状態で、Ａレコードを新しいファイルＡに出力し、ファイルＡから新しいＡレコードを入力する。

⑦　Ａキー、Ｂキーが共に最大値になると処理を終了し、そうでなければ③に戻る。

文字列処理

◆力任せの探索法

◇力任せの探索法（文字列の照合）とは

力任せの探索法は、テキスト文字列（長さｎ）の中にパターン文字列（長さｍ）と同じ長さの部分文字列をつくり、パターン文字列の対応する文字と順次比較し、全文字が一致すれば探索成功、比較途中で不一致になると、配列の次の要素に移動して、新しい部分文字列を作りパターン文字列と比較する。テキスト文字列のすべての要素と比較して不一致ならば探索不成功となる。

◇力任せの探索法の手順

① 一つのポインタｉをテキスト上に、もう一つのポインタｊをパターン上に置き、ｉおよびｊを１で初期化する。

② テキスト上にポインタｉを先頭に長さｍ（パターン文字列と同じ長さ）の部分文字列を作成し、部分文字列の要素の添字をｉ＋ｊ－１で表す。

③ 添字ｊを操作して、テキスト上の部分文字列とパターンの対応する要素を比較する。

④ 対応する要素の内容が一致するとき、ｊのインクリメントを行い、パターンの長さに等しい一致が生じると照合が得られる。

⑤ 対応する要素の内容が比較途中で不一致になるか、または照合が得られた場合、パターンの添字ｊをリセットし、テキスト上のポインタｉをインクリメントし、②に戻る。

⑥ ｉがテキスト上のｎ－ｍ＋１の要素を超えると文字列の探索

を終了する。

◇流れ図の内容

①　テキスト文字列のｉ番目の文字を先頭に長さｍの部分文字列を考え、テキストの部分文字列の要素を添字ｉ＋ｊ－１で表す。

②　ｉを１で、lastをｎ－ｍ＋１で、foundを０で初期化する。

③　ｊを１で初期化して照合ループに入る。

④　Ｔ(ｉ＋ｊ－１)＝Ｐ(ｊ)ならば、ｊ＞ｍになるまで、ｊ＋１→ｊの操作を繰り返す。

⑤　Ｔ(ｉ＋ｊ－１)≠Ｐ(ｊ)ならば、照合ループから脱出し、ｉ＋１→ｉを実行し、ｉを先頭とする新しい部分文字列を考え、③に戻る。

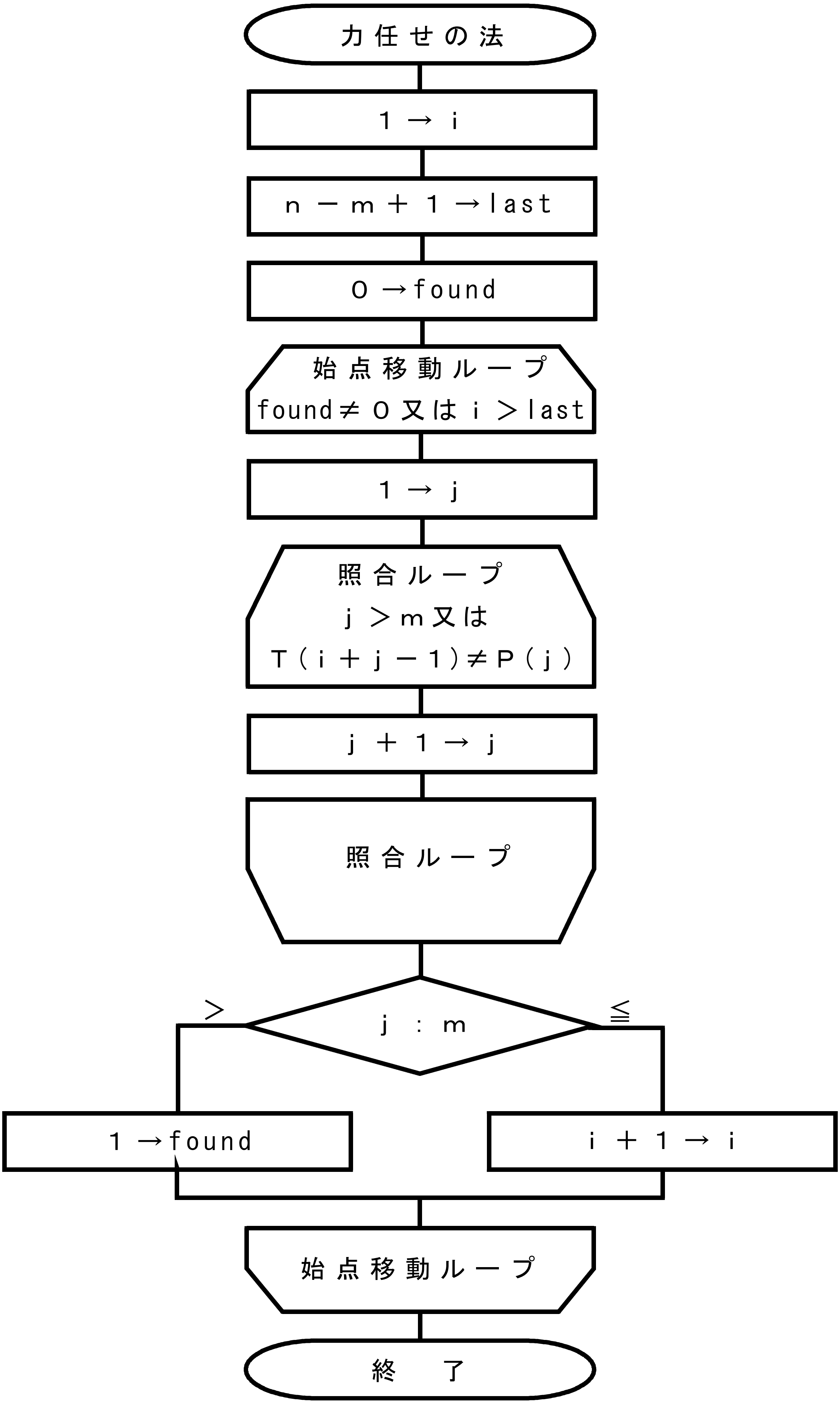
⑥　Ｔ(ｉ＋ｊ－１)＝Ｐ(ｊ)を繰り返し、ｊ＞ｍとなると照合成立となり、照合ループから脱出し、１→foundとなり、始点移動ループから脱出する。

⑦　始点移動が進み、ｉ＞ｎ－ｍ＋１となると、照合不成功で探索を終了する。

◇力任せの探索法の比較回数

テキストの長さをｎ、パターンの長さをｍとすると、力任せの探索法の最悪の場合の比較回数は(ｎ－ｍ＋１)×ｍとなる。

テキストの長さｎの値が、パターンの長さｍに比べて、大きい場合はｎｍとなる。


力任せの法
1 → i
n − m ＋ 1 →last
0 →found
始点移動ループ
found≠ 0 又は i ＞last
1 → j
照合ループ
j ＞m又は
T（i ＋ j − 1）≠P（j）
j ＋ 1 → j
照合ループ
＞
j ： m
≦
1 →found
i ＋ 1 → i
始点移動ループ
終　了

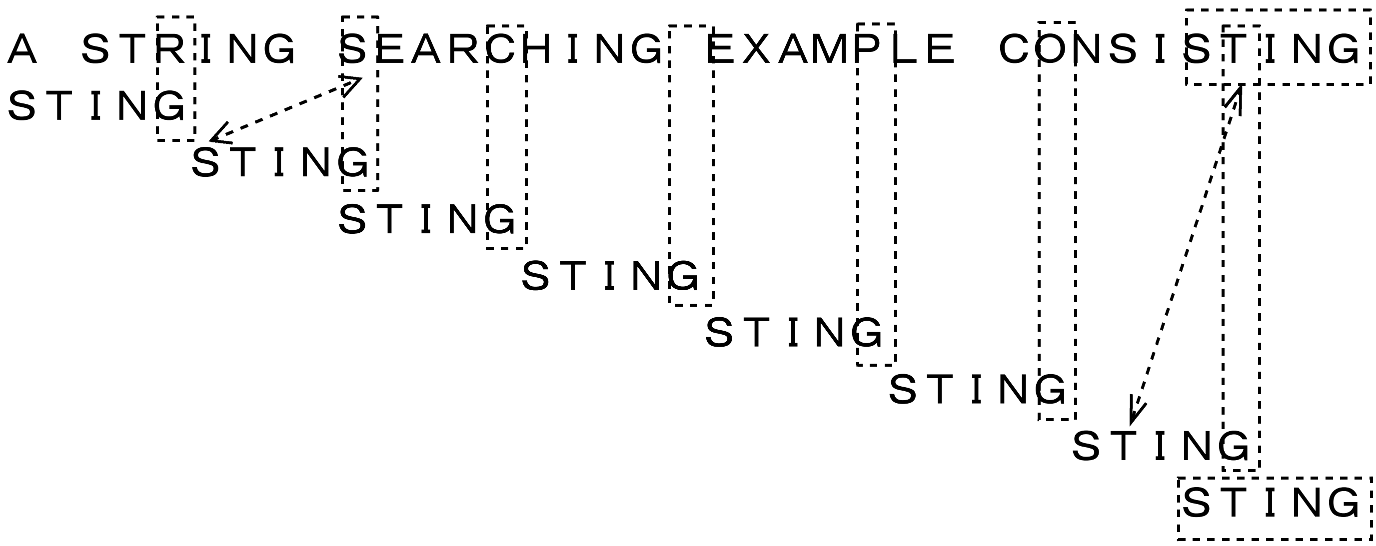
◆ボイヤ・ムーアの方法

◇ボイヤ・ムーアの文字列探索法

　パターンをテキストと照合させる時に右から左へポインタを動かす走査を取り入れることによってかなり速い文字列探索法を開発することができる。

　パターンだけでなく不一致を生じたテキスト文字も考慮してスキップの文字数を決定する。スキップの文字数は、テキストの各要素Ｔ（ｉ）に対して、移動距離として、配列ｄ（Ｔ（ｉ））に設定する。

◇パターン長５（ＳＴＩＮＧ）の場合の具体例



①　まず、最初にテキストの左端の５文字とパターンの５文字を比較する。

②　パターンの右端の文字とそれに対応するテキスト上の文字を比較し、一致すると、⑪の操作に移る。

③　一致しなければ、テキストとパターンの比較はパターンの右端の文字Ｇに対応するテキスト上の文字Ｒとパターンの各文字

を比較する。

④　テキスト上の文字Ｒがパターンの中に現れない文字であることがわかると、パターン全体を５文字(パターンの長さ)右に移動する。

⑤　次に、テキストの１０番目の文字Ｓとパターンの文字を比較する。文字Ｓとパターンの右端の文字が不一致でパターンの先頭の文字Ｓが一致する。

⑥　そこで、テキストの文字Ｓとパターンのｓが一致するまでパターンの文字列全体を右に４文字移動させる。

⑦　次に、テキスト上文字Ｃとパターン文字列を比較する。テキスト上文字Ｃがパターンの中に存在しないことが分かる。そこで、パターン文字列を右に５文字分移動し、同じことを繰り返す。

⑧　同様の移動を３回行うと、ＣＯＮＳＩＳＴＩＮＧのＴに到着する。

⑨　このテキスト上の文字Ｔがパターンの右端の文字と不一致であるが、パターンの中に存在するため、パターン文字列のＴがテキスト上の文字Ｔと一致するまでパターンを右に移動させる。

⑩　テキスト上の文字Ｇがパターンの右端の文字Ｇと一致する。

⑪　テキスト上の文字とパターン上の右端の文字が一致すると、それぞれの文字列の比較対象文字を一つ前の文字に移動し、二つの文字の比較を繰り返す。

⑫　テキストの文字列とパターンの文字列が完全に一致すると、探索成功になる。

⑬　テキスト文字列とパターン文字列の不一致が途中で発生する

と、パターン文字列の右端の文字に対応するテキスト上の文字と同じ文字がパターン上にあれば、二つの文字が一致するまで、パターン文字列を右に移動させる。

⑭　パターン文字列の右端の文字に対応するテキスト上の文字と同じ文字がパターン上になければ、パターン文字列の長さ分だけパターン文字列を右に移動させる。

◇文字列探索の要領

①　テキスト文字列の最初の比較文字は先頭からｍ番目とする。

②　パターン文字列の最初の比較文字は最後のｍ番目の文字からはじめる。

③　テキストの部分文字列の要素とパターン文字列の要素とが一致する間、ｉおよびｊのディクリメントを行い比較対象を変える。

④　Ｔ（ｉ）≠Ｐ（ｊ）となると、照合を中断して、照合ループから脱出する。

⑤　照合ループを脱出後、テキストの文字列の位置を最初のｍの位置に戻す。

⑥　照合開始時のテキストの文字列の要素Ｔ（ｉ）に対する移動距離を求める。

⑦　次の照合の開始点をｉ＋ｄ（Ｔ（ｉ））を求める。

⑧　照合を繰り返し、ｊ≦０となると、照合ループから脱出し、１→foundとして、更にｉ＋１→ｉを求めて、パターン文字列と一致するテキスト上の位置を決め、処理を終了する。

⑨　⑦で新しく求めた開始点を起点として、③～⑧の操作を繰り返す。

⑩　③～⑨の操作を繰り返して、ｉ＞ｎとなると、照合不成立で処理を終了する。

◇移動距離の計算要領

①　テキストの部分文字列とパターン文字列を比較する場合のテキスト文字列の最初の文字がパターン文字列の最後尾にあるか、または、パターン文字列の中に存在しない場合、パターン文字列の長さｍだけ移動する。

②　比較するテキスト文字列の最初の文字がパターン文字列の中にある場合、二つの文字の移動距離を求め、移動距離分だけ開始点を移動する。

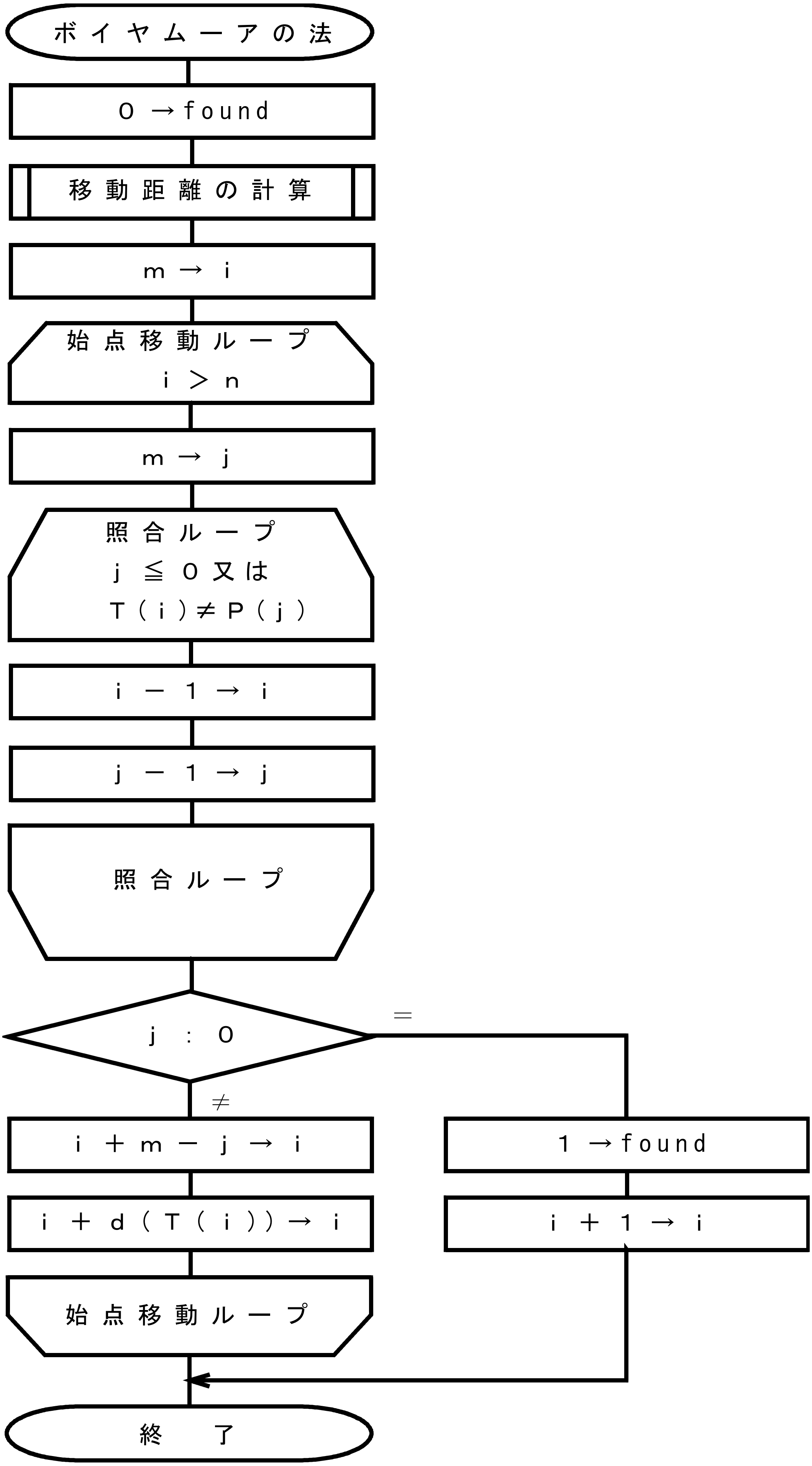
③　比較するテキスト文字列の最初の文字がパターン文字列の中に複数個ある場合、最後尾の文字までの移動距離を使用する。

◇ボイヤ・ムーアの比較回数

ボイヤ・ムーアの文字列探索における文字の比較回数の上限はＭ＋Ｎに比例する。

テキスト文字列の文字の種類がパターンの長さに比べて十分多ければ、比較回数はＮ／Ｍ回である。平均的な比較回数はＮ／Ｍとなる。

パターンの中に現れる文字は、アルファベットの一部にすぎないことが多い。このような場合、比較が行われるごとにＭ文字スキップされるようになる。このアルゴリズムは最悪の場合は線形となる。


ボイヤムーアの法
0 →found
移動距離の計算
m → i
始点移動ループ
i ＞ n
m → j
照合ループ
j ≦ 0 又は
T（i）≠P（j）
i − 1 → i
j − 1 → j
照合ループ
j : 0
=
≠
i ＋ m − j → i
i ＋ d（T（i））→ i
始点移動ループ
1 →found
i ＋ 1 → i
終　了

◆文字列の圧縮

◇文字列の圧縮

同一の文字が３文字以上連続する場合、次の３文字の様式に置き換える。

圧縮マーク＋連続する同じ文字の個数
＋連続する同じ文字の内容

圧縮後の文字列を同じ配列上に設定する場合と異なる配列上に設定する場合がある。

同じ配列上に設定する場合は、圧縮後の文字列の詰め合わせを実行しながら圧縮処理を行う必要がある。

◇文字列圧縮の手順

① 文字列の先頭から隣り合う２つの配列の要素を比較して、同一文字が連続する間、連続する同じ文字の個数を求める。

② 配列の隣り合う文字列が異なる場合に、文字の転送または文字列の圧縮を行う。

③ 連続する同じ文字数が２以下の場合は圧縮せずにそのまま転送する。

④ 連続する同じ文字数が３以上の場合、圧縮記号＋同じ文字数＋同じ文字の内容の３文字で表す。

⑤ 文字列の最後になると、終端記号を付加して終了する。

◇文字列圧縮の流れ図の内容

① 配列Ｓの添字ｉ、配列Ｐの添字ｊ、連続する同じ文字数を表す変数ｃを１で初期化する。

② 文字列の比較の繰り返しに入る。

終端文字が現れるまで文字列比較を繰り返す。終端文字が現れると、処理⑩に移る。

③ ｉをインクリメントし、着目する配列の要素を右隣に移動させる。

④ 配列Ｓの着目する要素ｉと一つ前の要素（ｉ－１）の文字を比較する。

２つの文字が等しい場合は処理⑬に移り、等しくない場合は処理⑤に移る。

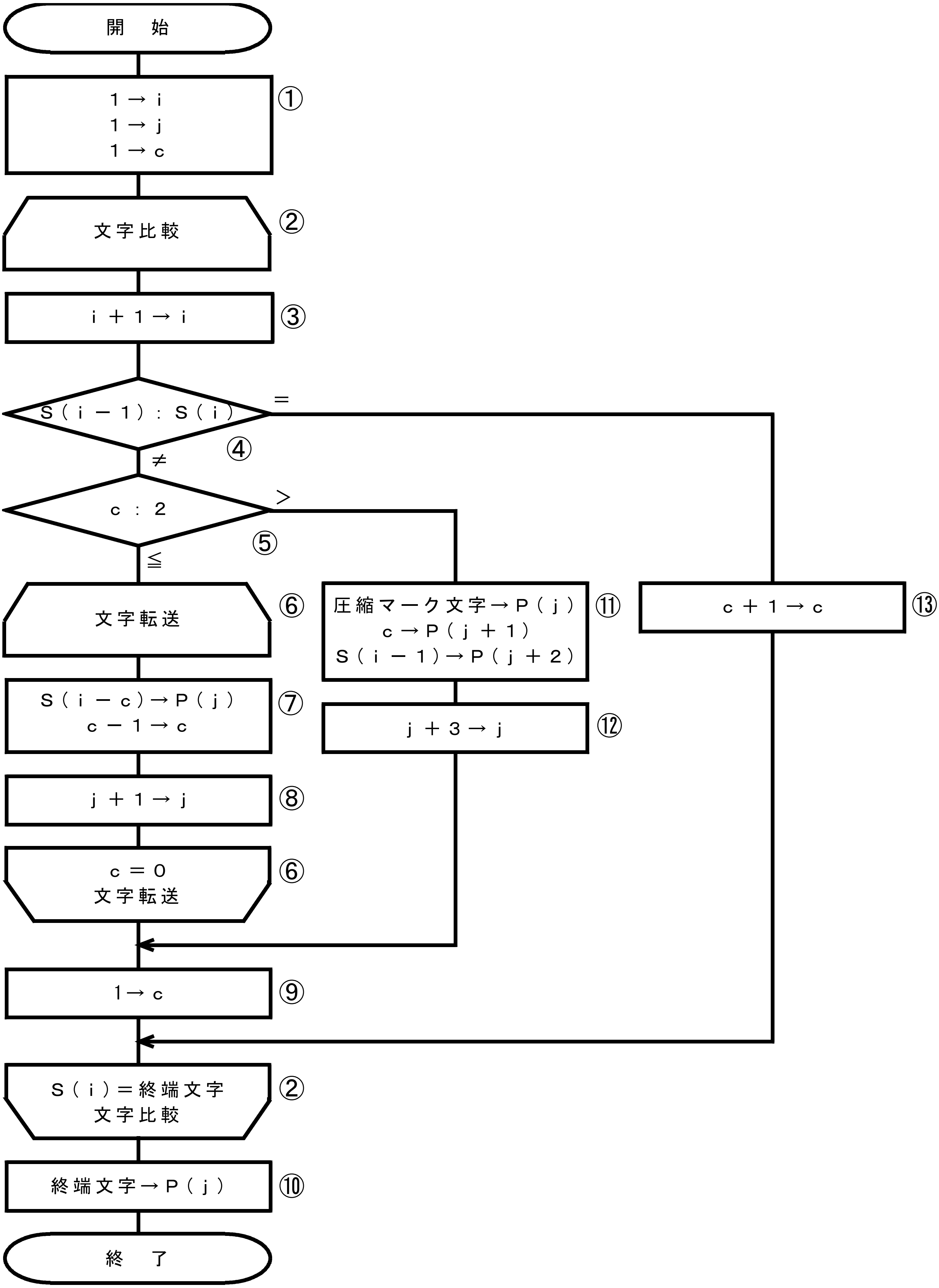
⑤ 連続する同じ文字の個数ｃと２を比較して、ｃ＞２の場合は処理⑪に移り、ｃ≦２の場合は処理⑥に移る。

⑥ 文字転送操作の繰り返しに入る。

ｃが２以下の場合、同じ文字数ｃを配列Ｓから配列Ｐに移動させる。従って、ｃが０になるまで転送操作⑦～⑧を繰り返す。

⑦ 配列Ｓの要素（ｉ－ｃ）の文字を配列Ｐの要素ｊに転送し、連続する同じ文字数ｃをディクリメントする。

⑧ 配列Ｐの要素の添字ｊをインクリメントし、次の転送位置に移動する。


開　始
1 → i
1 → j
1 → c
①
文字比較
②
i ＋ 1 → i
③
S（i － 1）：S（i）
④
＝
≠
c ： 2
⑤
＞
≦
文字転送
⑥
S（i － c）→ P（j）
c － 1 → c
⑦
j ＋ 1 → j
⑧
c ＝ 0
文字転送
⑥
圧縮マーク文字→ P（j）
c → P（j ＋ 1）
S（i － 1）→ P（j ＋ 2）
⑪
j ＋ 3 → j
⑫
c ＋ 1 → c
⑬
1→ c
⑨
S（i）＝終端文字
文字比較
②
終端文字→ P（j）
⑩
終　了

⑨　ｃ≦２の場合の全ての文字の転送、ｃ＞２の場合の文字列の圧縮の操作が完了すると、配列Ｓの連続する同じ文字の個数ｃを１で初期化する。

⑩　配列Ｐの要素の添字ｊに終端文字を格納する。

⑪　配列Ｓの文字列の同じ文字が３個以上連続する場合、３個の文字列に圧縮し、配列Ｐに格納する。

Ｐ（ｊ）に圧縮マーク文字
Ｐ（ｊ＋１）に同じ文字数のｃの値
Ｐ（ｊ＋２）に同じ文字の内容Ｓ（ｉ－１）

の文字を格納する。

⑫　配列Ｐの要素の次の挿入位置ｊ＋３を求める。

⑬　連続する同じ文字の個数ｃをインクリメントする。

◆文字列の置換

◇文字列の置換

テキスト文字列においてパターン文字列に一致する文字列を探索し、一致文字列を置換文字列に置き換える処理である。文字列の探索は、テキスト文字列上の部分文字列とパターン文字列の対応する要素を比較して、パターン文字列の大きさと同じ連続個数一致すれば探索成功となる。

処理結果を同じ配列上に設定する場合は、パターン文字列の大きさと置換文字列の大小関係によってテキスト文字列の挿入・削除の処理が必要になる。パターン文字列の大きさが置換文字列の大きさよりも大きい場合は詰め合わせが必要となり、削除の考え方を利用する。置換文字列の大きさがパターン文字列の大きさよりも大きい場合は空白が必要となり、挿入の考え方を利用する。

◇文字列置換の流れ図

大きさＮの配列Ａのテキスト文字列中に、長さＭの配列Ｂのパターン文字列が存在すると、その部分を長さＬの配列Ｃの文字列で置換する。

配列Ａの定義領域は無限として、置換は配列Ａ上で行う。また、ＭとＬの大小関係は不定である。

流れ図は次の主要部分で構成される。

① 配列Ａのテキスト文字列中の配列Ｂのパターンを探索する部分

② 配列Ａのテキスト文字列中に配列Ｃの文字列を挿入するための準備の部分

③　配列Ａのテキスト文字列中に配列Ｃの文字列を挿入する部分

◇文字列置換の流れ図の内容

①　ループＡの機能

配列Ａの部分文字列の先頭位置を１～Ｎ－Ｍ＋１まで走査する。

終了条件は、Ｉ＞Ｎ－Ｍ＋１となる。

②　ループＢの機能

配列Ａの部分文字列と配列Ｂのパターン文字列を比較する。

配列Ａの部分文字列は添字Ｉで始まる文字列で、文字列の操作は配列Ｂと同じＪを使用する。配列Ａの部分文字列の要素は（Ｉ＋Ｊ－１）で表すことができる。

配列Ａ（Ｉ＋Ｊ－１）、配列Ｂ（Ｊ）ともに、Ｊ＝１で初期化し、Ｊのインクリメントを使用して要素間を移動する。

対応する両者の文字列が不一致になると、ループＢから脱出するためのＳＷを使用する。最初は、ＳＷ＝０で初期化し、不一致になるとＳＷ＝１となる。

配列Ａ（Ｉ＋Ｊ－１）、配列Ｂ（Ｊ）の要素の比較が、Ｊ＞Ｍとなると、二つの文字列が一致する。

③　ループＣの機能

ループＢの脱出条件がＳＷ＝１の場合、グループＣを実行しない。スイッチングの役割を果たす

課題

配列Ａ （Ｎ）

Ｌ	Ｄ	Ｌ	Ｅ	Ａ	Ｓ	Ｕ	Ｂ

配列Ｂ （Ｍ）

Ｌ	Ｅ	Ａ

配列Ｃ （Ｌ）

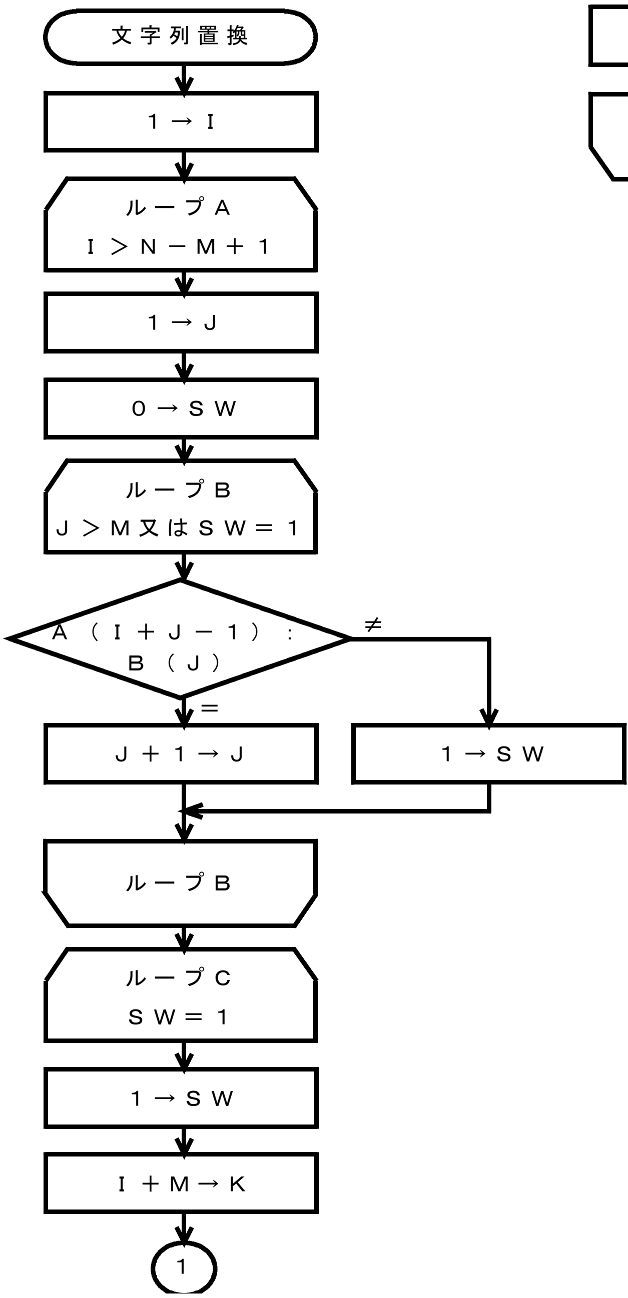
Ｐ	Ｕ	Ｓ	Ｈ

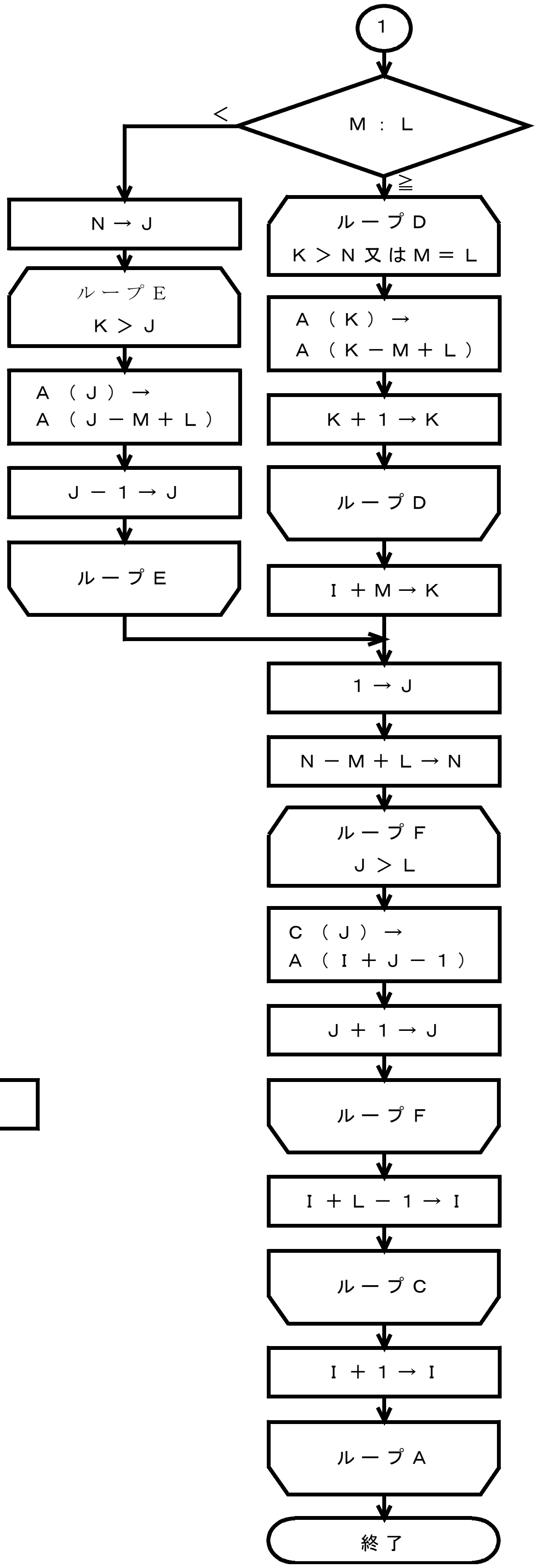
↓ 実行

実行後 配列Ａ

Ｌ	Ｄ	Ｐ	Ｕ	Ｓ	Ｈ	Ｓ	Ｕ	Ｂ

実行後の配列ＡのＮ □





④　ループＤの機能

Ｍ＞Ｌの場合、配列の削除の考え方を利用して置換領域の詰め合わせを行う。配列Ａの部分文字列の先頭位置Ｉから後方のＩ＋Ｍ－１の範囲は置換対象となるため、詰め合わせの最初の移動対象の位置はＩ＋Ｍ→Ｋとなる。

このＫを新たな添字として利用し詰め合わせの処理を行う。配列Ａの添字Ｉ＋ＭからＮまでの範囲をＭ－Ｌ前に詰める。Ａ（Ｋ）→Ａ（Ｋ－Ｍ＋Ｌ）を実行する。Ｍ＝Ｌの場合は詰め合わせの処理が不必要である。

⑤　ループＥの機能

Ｍ＜Ｌの場合、配列の挿入の考え方を利用して置換領域を後方に広げる。移動対象の添字ＪをＮで初期化して、配列Ａの後方からＩ＋Ｍの範囲をＬ－Ｍ後方にずらせる操作を行う。

Ａ（Ｊ）→Ａ（Ｊ－Ｍ＋Ｌ）、Ｊのディクリメント Ｊ－１→Ｊを実行する。終了条件は、④で定義したＫを使用してＪ＜Ｋとなる。

⑥　ループＦの機能

配列Ｃの文字列を配列Ａの中に置換する。Ｃ（Ｊ）→Ａ（Ｉ＋Ｊ－１）を実行する。

◆再帰アルゴリズム（リカーシブ）

◇リカーシブとは

　プログラムの中から自分自身を呼び出すことを再帰呼出という。自分自身を定義するのに自分自身よりも１次低い集合を用いる。その部分集合はより低次の部分定義を用いて定義することを繰り返して表現する。

　再帰呼出には再帰からの脱出口がなければならない。脱出口がない場合、再帰呼出は永遠に続くことになる。再帰呼出を行うときは、呼んだ自分自身から制御が戻ってきたときに、前の状態で引き続き実行できるように、実行アドレスや引数、使用していた変数などの情報を保管しておく。そのための保管領域として、後入れ先出しの特徴を持つスタックを使用する。

◇再帰呼び出しの具体例

　フィボナッチの数列

　1　1　2　3　5　8　13　21　34　55　89　…

　フィボナッチの数列の定義

$F_n = F_{n-1} + F_{n-2} \qquad n \geqq 3$

$F_1 = F_2 = 1 \qquad n = 1$または$2$

　フィボナッチの数列は再帰呼び出しを使用して処理ができる。定義した関数Ｆｎをスタックに格納し、Ｆｎの中で使用しているＦｎ－１を次に格納し、順次１次低い定義関数を格納していくと、最後に既知の値Ｆ２＝１、Ｆ１＝１にたどり着き、再帰呼び出しから脱出して、格納した順序とは逆に各関数の値を求めることができる。

◇スタックを利用した処理手順

① 次の各式を逐次スタックにＰＵＳＨする。

$$F_n = F_{n-1} + F_{n-2}$$
$$F_{n-1} = F_{n-2} + F_{n-3}$$
$$\vdots$$
$$F_4 = F_3 + F_2$$
$$F_3 = F_2 + F_1$$

② $F_2 = F_1 = 1$であるから、$F_2 + F_1 = 2$となり、再帰呼出しから脱出する。

③ スタックの各式をＰＯＰして値を求める。

$$F_3 = F_2 + F_1 = 2$$
$$F_4 = F_3 + F_2 = 3$$
$$\vdots$$
$$F_{n-1} = F_{n-2} + F_{n-3}$$
$$F_n = F_{n-1} + F_{n-2}$$

を利用してF_nを求めることができる。

◆バッカス記法

◇バッカス記法とは

　バッカス記法は言語を定義するための言語である。文章をどのようにして構成するかという構文規則を示している。

　バッカス記法の「：：＝」や「｜」を超記号と呼ぶ。「＜」と「＞」で囲まれたものを構文要素と呼び、これを構文則という。

◇構文則によって定義される文章

　＜文＞から始まる。構文要素をその右辺のものと置き換えて、構文要素を一つも含まなくなったものである。

◇表記法

①　＜数字＞：：＝０｜１｜２｜３｜４｜５｜６｜７｜８｜９

②　＜英字＞：：＝Ａ｜Ｂ｜Ｃ｜Ｄ｜Ｅ｜Ｆ｜Ｇ｜Ｈ｜Ｉ｜Ｊ｜Ｋ
｜Ｌ｜Ｍ｜Ｎ｜Ｏ｜Ｐ｜Ｑ｜Ｒ｜Ｓ｜Ｔ｜Ｕ
｜Ｖ｜Ｗ｜Ｘ｜Ｙ｜Ｚ

③　＜名前＞：：＝＜英字＞｜＜名前＞＜英字＞
｜＜名前＞＜数字＞

　①は「＜数字＞は０～９のどれかである」

　②は「＜英字＞はＡ～Ｚのどれかである」。「｜」は「または」と言う意味である。

　③の最初の＜名前＞：：＝＜英字＞は「英字は名前である」と読む。Ａという英字は名前として使うことができる。
　｜＜名前＞＜英字＞は「または、名前の次に英字をつけたもの

も名前である」と読む。Ａは名前であるから、その次に英字ＡをつけたものＡＡも名前である。ＡＡは名前であるから、その次に英字Ｙをつけたものの ＡＡＹも名前である。

｜＜名前＞＜数字＞は「または、名前の次に数字をつけたものも名前である」と読む。Ｂは名前であるから、その次に数字をつけたＢ３は名前である。Ｂ３は名前であるから、その次に数字９をつけたものＢ３９も名前である。名前は英字一つまたは英字で始まりその後に英字又は数字をいくつかつけたものである。

◇バッカス記法の具体例

日本語の文章は主部と述部からなり、主部は名詞と助詞からなり、述部は動詞からなるとする。

名詞は、私、君、犬、助詞は、は、が、も、動詞は、遊ぶ、泳ぐ、走る、があるものとし、「私は遊ぶ」という文章を考える。

＜文＞　::＝＜主部＞＜述部＞
＜主部＞::＝＜名詞＞＜助詞＞
＜述部＞::＝＜動詞＞
＜名詞＞::＝私｜君｜犬
＜助詞＞::＝は｜が｜も
＜動詞＞::＝遊ぶ｜泳ぐ｜走る

＜文＞::＝＜主部＞＜述部＞
｜＜名詞＞＜助詞＞＜述部＞
｜＜名詞＞＜助詞＞＜動詞＞
｜私＜助詞＞＜動詞＞
｜私は＜動詞＞｜私は遊ぶ

◆正規表現

◇正規表現とは

字句をパターンの集合で表す式で定義する表現方法である。文字列に対し、メタキャラクタといわれる特殊な記号を使用し、組み合わせて任意の文字列を表現する。

◇主な正規表現の例

メタキャラクタ	意味	例
^	行の先頭	＾Ａ行頭がＡの文字で始まる
＄	行の末尾	Ｚ＄行末がＺの文字で終わる
.	任意の１文字	Ｍ..Ｎ　Ｍで始まりＮで終わる４文字
?	任意の１文字	Ｍ??Ｎ　Ｍで始まりＮで終わる４文字
[]	[]内に含まれる任意の１文字	［Ａ－Ｚ］Ａ～Ｚの範囲の文字
＊	直前文字の０回以上の繰返し	ＡＢＸ＊　Ｘを０回以上繰返す
＋	直前文字の１回以上の繰返し	ＡＢＸ＋　Ｘを１回以上繰返す

◆分割統治法

◇分割統治とは

分割統治法は、解くべき問題を小規模な問題に分割し、各部分問題の解を結合することによって、全体の解を求めようとする解決法である。

分割というのは、対象とする変数の集合をいくつかに分けたり、定義領域を分けることであり、多くの場合、再帰的に反復して利用する。

クィックソートやマージソートは分割統治の考え方を利用したアルゴリズムであり、システム分析やプログラム設計にもこの考え方が用いられている。

モジュール分割は大きなプログラムを主たる部分とデータ入出力やデータチェック部分に分割し、各モジュールを個別に作成し、テストし、最後に各モジュールを結合して完全なプログラムを作成する。

アルゴリズム問題２

問１問題

　探索方法とその実行時間のオーダの正しい組合せはどれか。

　ここで，探索するデータ数をｎとし，ハッシュ値が衝突する（同じ値になる）確率は無視できるほど小さいものとする。また，実行時間のオーダがn^2であるとは，ｎ個のデータを処理する時間がcn^2（ｃは定数）で抑えられることをいう。

	２分探索	線形探索	ハッシュ探索
ア	$\log_2 n$	n	1
イ	$n \log_2 n$	n^2	1
ウ	n^2	1	n
エ	$n \log_2 n$	n	$\log_2 n$

◇問１解説

　探索法の計算量に関する問題である。

　線形探索を行う場合の比較回数の最大値はｎ回であり、線形探索のオーダはｎとなる。

　二分探索の平均比較回数は$[\log_2 n]$、最大比較回数はＫ＋１＝$[\log_2 n]$＋１となる。

　Ｎが十分に大きい場合、二分探索の比較回数は$[\log_2 n]$となり、二分探索のオーダは$[\log_2 n]$となる。

　ハッシュ法のオーダは通常、Ｏ(１)である。

以上の結果から、２分探索のオーダは$\log_2 n$、線形探索のオーダはｎ、ハッシュ探索のオーダは１となり、求める答えはアとなる。

次の表は線形探索、二分探索、ハッシュ法の計算量を示したものである。

	最大比較回数		
データ数	線形探索	二分探索	ハッシュ法
１０００	１０００	１０	1
１００００	１００００	１４	1
10^8	10^8	２７	1
N	N	$\log_2 N+1$	1

◇問１解答　ア

問２問題

コンピュータで連立一次方程式の解を求めるのに、未知数の個数の３乗に比例する計算時間がかかるとする。

あるコンピュータで１００元連立一次方程式の解を求めるのに２秒かかったとすると、その２倍の演算速度をもつコンピュータで１,０００ 元連立一次方程式の解を求めるには何秒かかるか。

ア　１０
イ　１００
ウ　１,０００
エ　１０,０００

◇問２解説

計算量を求める問題である。

計算時間は未知数の個数の３乗に比例するから、次のようにして求める。

１００元連立一次方程式は、１００個の未知数をもち、計算に２秒かかる。これを、２倍の演算速度で計算すると、１秒になる。１０００元の連立方程式は未知数を１０００個もつため次の関係式が成り立つ。

$$(100)^3 : (1000)^3 = 1 : X$$
$$10^6 : 10^9 = 1 : X$$
$$\therefore X = 1000$$

１０００秒をかかる。求める答えはウとなる。

◇問２解答　ウ

問３問題

昇順に整列されたｎ個のデータが配列に格納されている。探索したい値を２分探索法で探索するときの，およその比較回数を求める式はどれか。

ア　$\log_2 n$
イ　$(\log_2 n + 1)/2$
ウ　n
エ　n^2

◇問３解説

二分探索法の比較回数の問題である。

データ数ｎの場合、二分探索の平均比較回数をＫとすると、次の式が成り立つ

$$2^K \leqq n < 2^{K+1}$$

両辺の２を底とする対数をとると

$$\log_2 2^k \leqq \log_2 n < \log_2 2^{K+1}$$
$$K \leqq \log_2 n < K + 1$$
$$K = [\log_2 n]$$

平均比較回数は$[\log_2 n]$となる。求める答えはアとなる。

◇問３解答　ア

問４問題

２分探索において、整列されているデータ数が４倍になると、最大探索回数はどうなるか。

ア　１回増える
イ　２回増える
ウ　約２倍になる
エ　約４倍になる

◇問４解説

２分探索のデータ数と探索回数の関係に関する問題である。

データ数をＮとすると、探索回数は$\log_2 N$に比例する。最大探索回数は$\log_2 N + 1$となる。

データ教が４倍になると、最大探索回数は次のようになる。

$$
\begin{aligned}
\log_2 4N + 1 &= \log_2 4 + \log_2 N + 1 \\
&= 2 + \log_2 N + 1 \\
&= 3 + \log_2 N
\end{aligned}
$$

となる。従って、２回探索回教が増加する。求める答えはイとなる。

◇問４解答　イ

問５問題

　データを降順に並べた線形リストを２分探索法で探索するとき、３回目までの比較で探索を終了することができる要素の最大個数はどれか。

ア　４
イ　５
ウ　７
エ　１５

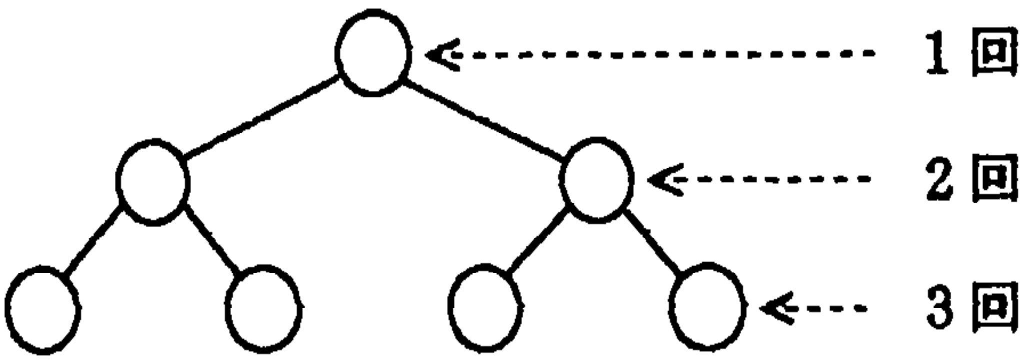
◇問５解説

　２分探索におけるデータの個数と比較回数に関する問題である。

　平均比較回数は$\log_2 N$に比例する。

　２分探索木の考え方を利用して、高さが２の場合について、最大個数を求めればよい。

　２分探索法では、中央値を比較し、探索値が中央値よりも大きい場合は左部分木へ、小さい場合には右分木に進み、同様の比較を繰り返す。



図は、探索３回までの２分木を表したものである。要素の最大個数は７個になる。求める答えはウとなる。

２分木の階層数Ｋと要素の最大数Ｎとの間には、次式が成立する。

$$N = 2^{K} - 1$$

Ｋ＝３とすると、

$$N = 2^{3} - 1 = 7$$

◇問５解答　ウ

問６問題

　次のような数値が格納された配列から数値４を二分探索法で検索する場合、比較回数はどれか。

　　2、4、6、7、8、10、11、12、15、16、18

ア　２
イ　３
ウ　４
エ　６

◇問６解説

　２分探索の比較回数を求める問題である。

添字	1	2	3	4	5	6	7	8	9	10	11
	2	4	6	7	8	10	11	12	15	16	18

　１回目の中央値は添字６で１０と４との比較になる。次の範囲は１～５で、２回目の中央値は添字３で６と４との比較になる。次の範囲は１～２で、３回目の中央値は添字１で２と４の比較になる。４回目の範囲は２～２で添字２の４と４の比較で一致する。
　求める答えはウとなる。

◇問６解答　ウ

問７問題

２，０００ 個の相異なる要素が、キーの昇順に整列された表がある。外部から入力したキーによってこの表を２分探索して、該当するキーの要素を取り出す。このときのキーの比較回数は最大何回か。

ただし、該当するキーは必ず表中にあるものとする。

ア　１０
イ　１１
ウ　１２
エ　１３

◇問７解説

探索データが存在する場合の２分探索法の最大比較回数を求める問題である。

最大比較回数をｋとすると、要素数Ｎとの間に次の条件が成立する。

$$2^{k-1} \leqq N < 2^{k}$$

Ｎ＝２０００として、ｋの値を求める。

$$2^{k-1} \leqq 2 \times 10^{3} < 2^{k}$$

$2^{k-1} \leqq 2 \times 10^{3}$から

$$k - 1 \leqq \log_2 2 + \log_2 10^{3}$$

$$
\begin{aligned}
k &\leqq 1+1+3/\log_{10}2 \\
&= 2+3/0.301 \\
&= 2+9.967 \\
&= 11.967 \cdots ①
\end{aligned}
$$

$2 \times 10^3 < 2^k$

$$
\begin{aligned}
k &> \log_2 2+\log_2 10^3 \\
&= 1+3/0.301 \\
&= 1+9.967 \\
&= 10.967 \cdots ②
\end{aligned}
$$

①、②の式から　$k=11$となる。求める答えはイとなる。

◇問７解答　イ

問８問題

昇順に整列された１０００個の整数の配列から整数ｘに一致する要素を探す。逐次探索の場合と２分探索の場合との平均比較数の比（逐次探索：２分探索）はどれに近いか。

ここで、整数ｘは配列中に存在するすべての整数の値を等しい確率でとる。また、$\log_2$１０００＝１０としてよい。

ア　１００：１
イ　５０：１
ウ　２５：１
エ　１０：１

◇問８解説

２分探索法と線形探索法の計算量の比較の問題である。

それぞれの比較回数は、データ数をＮとすると、

２分探索の平均比較回数は　$\log_2$Ｎ
逐次探索は　Ｎ／２

データ数１０００の場合の（線形探索：２分探索）の値を求める、

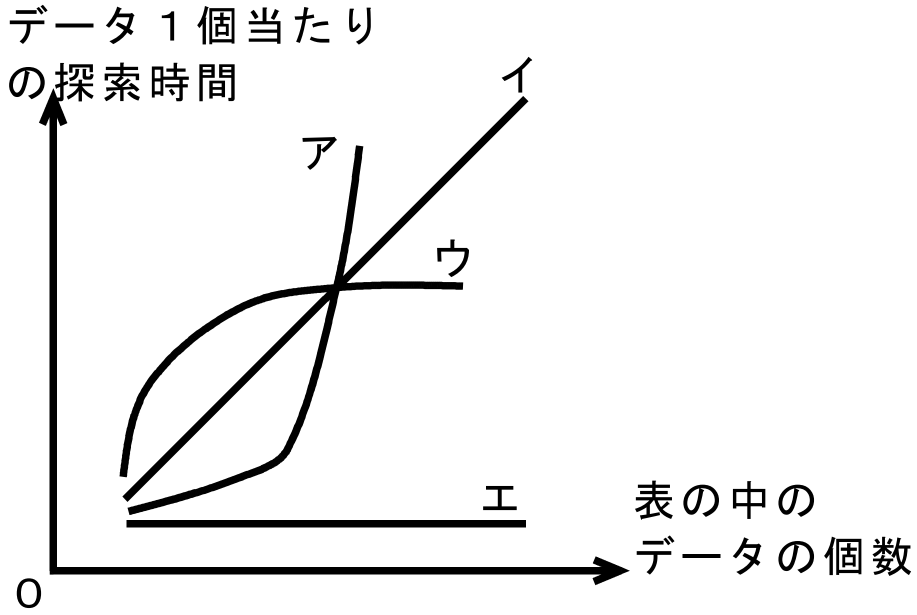
１０００／２：$\log_2$１０００
＝５００：３／０.３０１
＝５００：１０
＝５０：１

求める答えはイとなる。

◇問８解答　イ

問９問題

　次のグラフのうち、ハッシュ表探索の探索時間の特徴を表すものはどれか。ここで、シノニムは発生しないものとする。



◇問９解説

　ハッシュ法と探索時間のグラフに関する問題である。

　シノニムの発生しないハッシュ法は計算時間で決まるためデータ数に関係なく、探索時間は一定である。求める答えはエとなる。

◇問９解答　エ

問10問題

相異なるｎ個のデータが昇順に整列された表がある。この表を１ブロックｍ個に分割し，各ブロックの最後尾のデータだけ線形探索することによって，目的のデータの存在するブロックを探し出す。

次に当該ブロック内を線形探索して目的のデータを探し出す。このときの平均探索回数はどれか。

ここで，ｍ＜ｎとし，目的のデータは必ず表の中に存在するものとする。

ア $\frac{n}{m}$　　イ $\frac{n}{2m}$　　ウ $m+\frac{n}{m}$　　エ $\frac{m}{2}+\frac{n}{2m}$

◇問10解説

線形探索の計算量の問題である。

ｎ個のデータを１ブロックｍ個に分割すると、ブロック数は最大ｎ／ｍ＋１個となる。

探索するデータがどのブロックに属するかを調べるために、各ブロックの最後尾の値と比較する線形探索を用いると平均比較回数はｎ／２ｍ回となる。

各ブロックの中で、線形探索を実行するため平均探索回数はｍ／２となる。

全平均比較回数は、

ｎ／２ｍ＋ｍ／２

となる。求める答えはエとなる。

この種の問題としては、探索成功確率と組み合わせた問題、２分探索と線形探索を組み合わせた場合の平均探索回数を求める問題などがある。

◇問10解答　エ

問11問題

　未整列の配列Ａ［ｉ］（ｉ＝１，２，…，ｎ）を，次のアルゴリズムで整列する。要素同士の比較回数のオーダを表す式はどれか。

〔アルゴリズム〕

（１）Ａ［１］～Ａ［ｎ］の中から最小の要素を探し，それをＡ［１］と交換する。

（２）Ａ［２］～Ａ［ｎ］の中から最小の要素を探し，それをＡ［２］と交換する。

（３）同様に，範囲を狭めながら処理を繰り返す。

ア　$O(\log_2 n)$
イ　$O(n)$
ウ　$O(n \log_2 n)$
エ　$O(n^2)$

◇問11解説

　選択ソートに関する問題である。

　最初の１回の比較回数はデータ数がＮ個であるからＮ－１回となる。

　次の１回の比較回数はデータ数がＮ－１個であるからＮ－１となる。

　同様に最後の２個になるまで繰り返すと、Ｎ－１回実行することになる。

従って、総比較回数は次のようになる。

$$
\begin{aligned}
&(N-1)+(N-2)+(n-3)+\cdots+2+1 \\
&=N(N-1)/2 \\
&=(N^2-N)/2
\end{aligned}
$$

比較回数のオーダはO（n^2）となる。求める答えはエとなる。

◇問11解答　エ

問12問題

あるコンピュータで １,０００個のデータをバブルソートを用いて整列するのに、１秒かかった。同様のデータ１００,０００個ではおよそ何秒かかるか。

但し、バブルソートの手間はデータ数の２乗に比例する。

ア　１０
イ　１００
ウ　１,０００
エ　１０,０００

◇問12解説

バブルソートの計算量に関する問題である。

バブルソートの計算量は、データ数の２乗に比例する。

１０００個のデータの操作時間が１秒であるから、１００００００個の場合の操作時間をＴとすると、次の式が成り立つ。

$$(10^3)^2 : (10^5)^2 = 1 : T$$
$$10^6 : 10^{10} = 1 : T$$
$$1 : 10^4 = 1 : T$$

従って、Ｔ＝１００００（秒）となる。求める答えはエである。

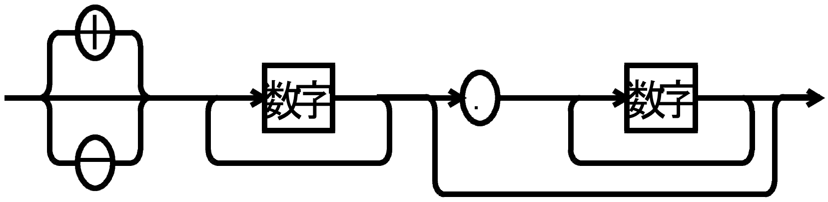
◇問12解答　エ

問13問題

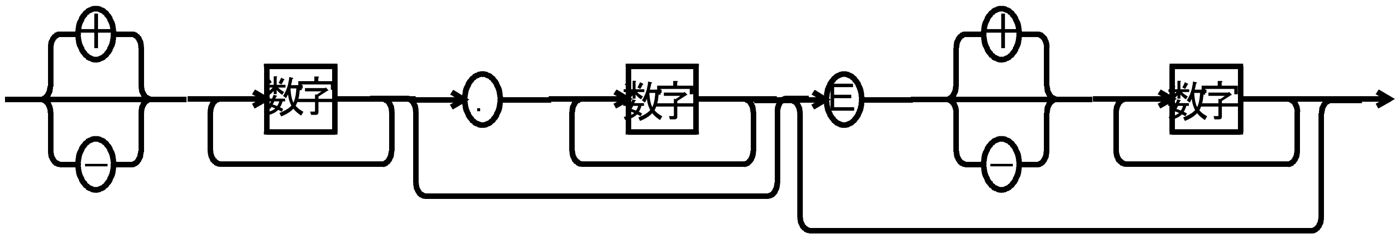
　第１図のような記述法による構文図を考える。－１、２１．５、＋５．２３などの表現は、この構文図の規定に合致する。

　この記述法に従うとき、第２図の規定に合致する数値表現はどれか。

第1図



第2図



ア　－５２．３Ｅ０５
イ　．５８Ｅ２
ウ　＋．９０Ｅ１１
エ　３．４５Ｅ

◇問13解説

　構文図に関する問題である。

　第１図を利用して基本的な考え方を理解し、その考え方を利用して、第２図の構文図の内容を読み取る問題である。

第１図の内容から次のことが分かる。

① 数字の前には、＋－の記号または何も付加されない。

② 数字は１個以上の並びになる。

③ 数字列の後の小数点はつく場合とつかない場合がある。

④ 小数点が付加されると、その後には１個以上の数字列がくる。

第２図では、指数表示のＥの項目が付加される。

アは構文図の規定に合致している。求める答えはアとなる。
イは小数点の前に数字がない。
ウは＋と小数点の間に数字がない。
エはＥの後に数字がない。

◇問13解答　ア

問14問題

　数値に関する構文が次のとおり定義されているとき，＜数値＞として扱われるものはどれか。

　＜数値＞：：＝＜数字列＞｜
　　　　　　　　＜数字列＞Ｅ＜数字列＞｜
　　　　　　　　＜数字列＞Ｅ＜符号＞＜数字列＞
　＜数字列＞：：＝＜数字＞｜＜数字列＞＜数字＞
　＜数字＞：：＝０｜１｜２｜３｜４｜５｜
　　　　　　　　６｜７｜８｜９
　＜符号＞：：＝＋｜－

ア　－１２
イ　１２Ｅ－１０
ウ　＋１２Ｅ－１０
エ　＋１２ＥＩ０

◇問14解説

　構文に関する問題である。

　ア、ウ、エは先頭に符号があるため定義された構文ではない。
　イは＜数字列＞Ｅ＜符号＞＜数字列＞の構文を満たすため答えとなる。求める答えはイとなる。

◇問14解答　イ

問15問題

　次に示すのは，ＢＮＦで記述されたあるプログラム言語の構文の一部である。＜パラメタ指定＞として正しいものはどれか。

＜パラメタ指定＞
　　：：＝＜パラメタ＞｜
　　　　（＜パラメタ指定＞，＜パラメタ＞）
＜パラメタ＞
　　：：＝＜英字＞｜＜パラメタ＞＜英字＞
＜英字＞：：＝ａ｜ｂ｜ｃ｜…｜ｘ｜ｙ｜ｚ

ア　（ａｂｃ）
イ　((ａｂｃ，ｄｅｆ))
ウ　（ａｂｃ，（ｄｅｆ))
エ　((ａｂｃ，ｄｅｆ)，ｇｈｉ）

◇問15解説

　バッカス記法に関する問題である。

　パラメータの指定は、連続する英文字または、連続する英文字が２個ある場合は、２つの文字列を（　）で括り、（　）内の２つの文字列はカンマで区切る。文字列が３個ある場合は、そのうちの２つを（　）で括り、その文字列のグループともう一つの文字列を（　）で括る。

　アの場合、文字列は１つであるから（　）は不要である。
　イの場合、（　）は２つもいらない。１つのみでよい。
　ウの場合、ｄｅｆの（　）は不要である。
　エの構文は適切である。求める答えはエとなる。

◇問15解答　エ

問16問題

　次のＢＮＦで定義されるビット列Ｓであるものはどれか。

　　　＜Ｓ＞∷＝０１｜０＜Ｓ＞１

ア　０００１１１
イ　０１００１０
ウ　０１０１０１
エ　０１１１１１

◇問16解説

　ＢＮＦに関する問題である。

　０１、または０＜Ｓ＞１であるから、０と１の間に０１が挟まるビットパターンになる。

　アは０１の間に０１が挟まるビットパターンである。求める答えはアとなる。
　イは０と０の間に０１が挟まるパターンになっており正しくない。
　ウは０と１の間に１０が挟まるパターンであり正しくない。
　エは０と１の間に１１が挟まるパターンであり正しくない。

◇問16解答　ア

問17問題

　次の表は、入力文字列を検査するための状態遷移表である。この検査では、初期状態をａとして、文字列の入力中に状態がｅになれば不合格とする。解答群で示される文字列のうち、この検査で不合格となるものはどれか。
　ここで、解答群中の△は空白を示す。

		入力文字				
		空白	数字	符号	小数点	その他
現状の状態	a	a	b	c	d	e
	b	a	b	e	d	e
	c	e	b	e	d	e
	d	a	e	e	e	e

ア　＋００１０
イ　－１
ウ　１２．２
エ　９．△

◇問17解説

　状態遷移表を利用した文字列検査に関する問題である。

　ア～エについて、遷移を実行すると次のようになる。

　　アの場合、ａ→ｃ→ｂ→ｂ→ｂ→ｂ
　　イの場合、ａ→ｃ→ｂ
　　ウの場合、ａ→ｂ→ｂ→ｄ→ｅ、不合格になる。求める答えはウとなる。
　　エの場合、ａ→ｂ→ｄ→ａ

◇問17解答　ウ

問18問題

正規表現[A－Z]＋[0－9]*が表現する文字列の集合の要素となるものはどれか。

ここで，正規表現は次の規則に従う。

[A－Z]は，英字１文字を表す。
[0－9]は，数字１文字を表す。
＊ は，直前の正規表現の０回以上の繰返しを表す。
+ は，直前の正規表現の１回以上の繰返しを表す。

ア　456789
イ　ABC99*
ウ　ABC+99
エ　ABCDEF

◇問18解説

正規表現に関する問題である。

アの場合、英字１文字の１回以上の表現が不足している。
イの場合、＊の部分の処理が不十分である。
ウの場合、＋の部分の処理が不十分である。
エの場合、数字が０回の場合に相当する。正しい。求める答えはエとなる。

◇問18解答　エ

問19問題

　次の方法によって，データに検査数字（チェックディジット）を付加する。データにエラーが含まれていない場合，N_2＝７，N_3＝６，N_4＝２，Ｃ＝４のとき，N_1の値は幾らか。

元のデータ：$N_1N_2N_3N_4$

検査数字：Ｃ＝mod((
N_1×１＋N_2×２
＋N_3×３＋N_4×４)，１０)

ここで，mod(ｘ，１０)の値は，ｘを１０で割った余り

検査数字を付加したデータ：$N_1N_2N_3N_4$Ｃ

ア　0
イ　2
ウ　4
エ　6

◇問19解説

　チェックディジットに関する問題である。

　N_1の値をＸとすると、次の式が成立する。

mod(Ｘ＋７×２＋６×３＋２×４、１０)＝４

Ｘ＋７×２＋６×３＋２×４＋(１０－４)
＝Ｘ＋１４＋１８＋８＋６
＝Ｘ＋４６

Ｘ＋４６が１０の倍数になるＸの値を求めればよい。

　アの場合は４６、イの場合は４８、ウの場合は５０、エの場合は５２となる。求める答えはウとなる。

◇問19解答　ウ

問20問題

　試験の合否を判定する次の決定表から読み取れるものはどれか。ここで，試験は労務管理，経理及び英語の３科目で構成され，それぞれの満点は１００とする。

業務経験年数≧５	Y	Y	Y	N
３科目合計得点≧２６０	Y	Y	N	－
英語得点≧９０	Y	N	－	－
合格	X	－	－	－
仮合格	－	X	－	－
不合格	－	－	X	X

ア　英語の得点が９０以上の者は，仮合格か合格になる。
イ　英語の得点が９０未満の者は，不合格になる。
ウ　業務経験年数が５以上の者は，仮合格か合格になる。
エ　経理の得点が６０未満の者は，不合格になる。

◇問20解説

　決定表に関する問題である。

　アの英語の得点が９０以上の場合は合格である。
　イの英語の得点が９０未満の場合は仮合格になる。
　ウの業務経験年数５年以上は合格、仮合格、不合格になる。
　エの経理の得点が６０未満は不合格になる。３科目の合計点が２６０以上が合格であるから、経理以外の２科目が１００点で、経理が６０未満だと不合格になる。求める答えはエとなる。

◇問20解答　エ

問21問題

アルファベット３文字で構成されるキーがある。次の式によってハッシュ値ｈを決めるとき，キー“ＳＥＰ”と衝突するのはどれか。ここで，ａmodｂは，ａをｂで割った余りを表す。

ｈ＝(キーの各アルファベットの順位の総和)mod２７

アルファベット	順位	アルファベット	順位	アルファベット	順位
A	1	J	１０	S	１９
B	2	K	１１	T	２０
C	3	L	１２	U	２１
D	4	M	１３	V	２２
E	5	N	１４	W	２３
F	6	O	１５	X	２４
G	7	P	１６	Y	２５
H	8	Q	１７	Z	２６
I	9	R	１８		

ア　ＡＰＲ
イ　ＦＥＢ
ウ　ＪＡＮ
エ　ＮＯＶ

◇問21解説　イ

ハッシュ法の衝突に関する問題である。

ＳＥＰ＝１９＋５＋１６＝４０
４０　mod　２７＝１３

ア～エについてハッシュ値を求める。

アのＡＰＲ＝１＋１６＋１８＝３５
　３５　mod　２７＝８

イのＦＥＢ＝６＋５＋２＝１３
　１３　mod　２７＝１３

ウのＪＡＮ＝１０＋１＋１４＝２５
　２５　mod　２７＝２５

エのＮＯＶ＝１４＋１５＋２２＝５１
　５１　mod　２７＝２４

ＳＥＰと衝突するのはＦＥＢであり、求める答えはイとなる。

◇問21解答　イ

問22問題

次のように定義されている再帰関数がある。ｘ＝５０のときの値はいくらか。

```
F（x）：＝if x ＞ 50 then x － 5
                     else F（F（x ＋ 6））
```

ア　４５
イ　４６
ウ　５１
エ　６２

◇問22解説

再帰関数に関する問題である。

ｘ＝５０の場合

Ｆ（Ｆ（５０＋６））＝Ｆ（Ｆ（５６））
　　　　　　　　　＝Ｆ（５６－５）
　　　　　　　　　＝Ｆ（５１）
　　　　　　　　　＝５１－５＝４６

求める答えはイとなる。

◇問22解答　イ

問23問題

　次の関数ｇ（ｘ）の定義に従ってｇ（４）を再帰的に求めるとき、必要な加算の回数は幾らか。

　　ｇ（ｘ）＝if　ｘ＜２　then　１
　　　　　　　　　　　　　else　ｇ（ｘ－１）＋ｇ（ｘ－２）

ア　３
イ　４
ウ　５
エ　７

◇問23解説

　再帰関数に関する問題である。

　ｘ＝４の場合

$$
\begin{aligned}
g(4) &= g(4-1)+g(4-2)\\
&= g(3)+g(2)\\
&= g(3-1)+g(3-2)\\
&\qquad\quad +g(2-1)+g(2-2)\\
&= g(2)+g(1)+g(1)+g(0)\\
&= g(2-1)\underline{+}g(2-2)\\
&\qquad\quad \underline{+}g(1)\underline{+}g(1)\underline{+}g(0)\\
&= g(1)+g(0)+3=1+1+3\\
&= 5
\end{aligned}
$$

　必要な加算の回数は下線の箇所（赤色の＋記号）の４回である。求める答えはイとなる。

◇問23解答　イ

問24問題

自分自身を呼び出すことができるプログラムを、再帰プログラムという。

いま自然数ｎに対する階乗

ｎ！＝ｎ×（ｎ－１）×（ｎ－２）×…×２×１

を求めるために再帰プログラムを使用するとき、このプログラムの処理順序として正しいものはどれか。

ただし、ｎはこのプログラムに渡される自然数が入った引数、Ｆはこのプログラムでの処理結果を返すための引数である。

①　ｎ－１を計算して結果をｎに代入する。
②　ｎがゼロならＦ＝１として、プログラムを終了する。
③　ｎをスタックへ退避させる。
④　Ｆ×ｎを計算して結果をＦに格納する。
⑤　自分自身を呼び出す。
⑥　退避してあったｎをスタックから取り出す。

ア　②→①→③→⑤→④→⑥
イ　②→①→③→⑤→⑥→④
ウ　②→③→①→⑤→④→⑥
エ　②→③→①→⑤→⑥→④

◇問24解説

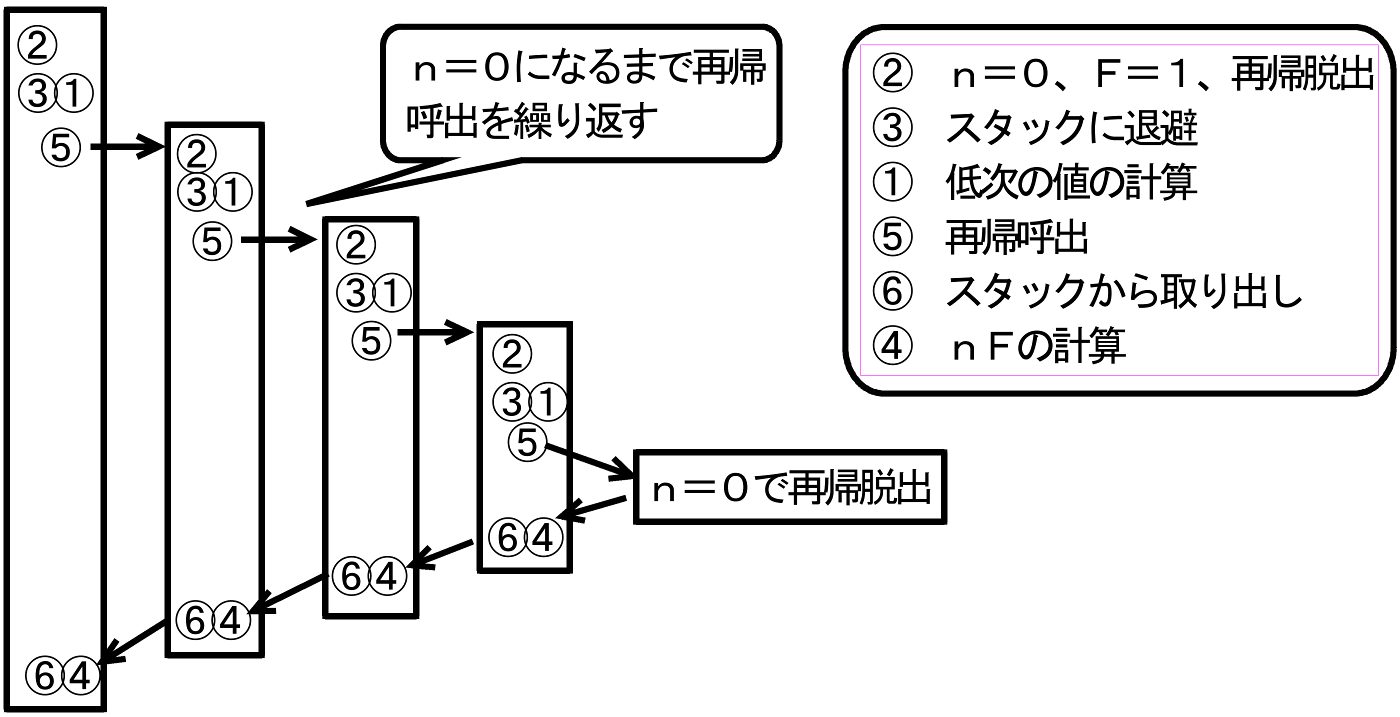
再帰プログラムに関する問題である。

再帰呼出には再帰からの脱出口がなければならない。脱出口がない場合、再帰呼出は永遠に続くことになる。

再帰呼出を行うときは、制御が戻ってきたときに、前の状態で引き続き実行できるように、実行アドレスや引数、使用していた変数などの情報を保管しておく。そのための保管領域として、後入れ先出しの特徴を持つスタックを使用する。

ｎ！の計算を再帰呼出を利用した手順は次のようになる。

① ｎが０ならば、Ｆ＝１として、再帰呼出から脱出する
② ｎ！＝ｎ（ｎ－１）！で、ｎをスタックに退避させる。
③ 次に、（ｎ－１）！を計算するために、再帰呼出を行う。処理が①に戻る。
④ ｎ＝０になるまで、ｎ－２、ｎ－３、…、２、１の処理を繰り返す。
⑤ ｎ＝０となると、Ｆ＝１となり、再帰呼出から脱出する。
⑥ ｎをスタックから取り出す。
⑦ Ｆ×ｎを計算して、結果をＦに代入する。



処理順序は、②③①⑤⑥④となり、求める答えはエとなる。

◇問24解答　エ

問25問題

整数Ａを整数Ｂで割って余りを得るための関数mod（Ａ、Ｂ）が次のように定義されているとき、関数呼出によって得られる値として正しいものはどれか。

［定義］
mod（Ａ、Ｂ）は、除数Ｂと同じ符号をとり、その絶対値はＢの絶対値より小さい、適切な整数Ｎを選ぶことによって、

Ａ＝Ｂ×Ｎ＋mod（Ａ、Ｂ）

を満足する。

ア　mod（１１，５）＝２
イ　mod（１１，－５）＝－１
ウ　mod（１２，－５）＝－３
エ　mod（－１２，５）＝２

問25解説

余りを求める関数mod（Ａ，Ｂ）に関する問題である。

次の点に注意して処理を検討する必要がある。

①　余りは除数Ｂと同じ符号である。

②　余りの絶対値は除数Ｂの絶対値より小さい。

③　Ａ、Ｂ、余りの間には次の関係が成立する。

Ａ＝Ｂ×Ｎ＋mod（Ａ，Ｂ）
但し、Ｎは整数である

この条件を利用して、解答群のア～エについて検討する。

アは１１÷５の余りを求める問題で、mod（１１，５）＝１であるから正しくない。

イは１１÷（－５）の余りを求める問題で、①、②の条件から商は３、余りは－４となる。正しくない。

ウは１２÷（－５）の余りで、商は３、余りは－３となる。正しい。求める答えはウとなる。

エは（－１２）÷５の余りを求める問題である。①、②の条件から商は３で、余りは３となる。Ａ＝５×（－３）＋３＝－１２。正しくない。

◇問25解答　ウ

問26問題

　ｐを２以上の整数とする。任意の整数ｎに対して，

　　　　ｎ＝ｋｐ＋ｍ（０≦ｍ＜ｐ）

を満たす整数ｋとｍが一意に存在する。このｍをｎのｐによる剰余といい，ｎ mod ｐで表す。

　（－１００００）mod ３２７６８に等しくなるものはどれか。

ア　－（１００００ mod ３２７６８）
イ　（－２２７６８）mod ３２７６８
ウ　１００００ mod ３２７６８
エ　２２７６８ mod ３２７６８

◇問26解説

　剰余に関する問題である。

　ｎ＝ｋｐ＋ｍで、０≦ｍ＜ｐであるから、ｎが負の場合、ｋが負で、ｍの値が正になる答えとなる。

　ｎの絶対値がｐよりも小さい場合は次のようになる。

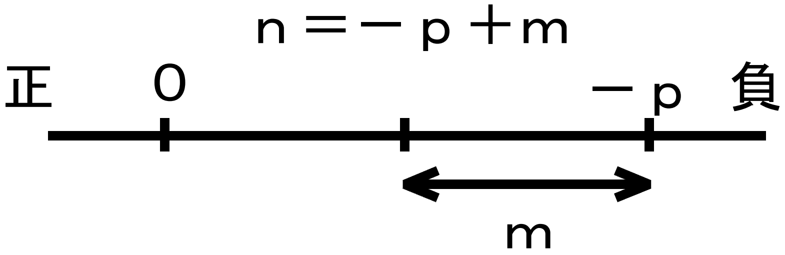
　ｎ＜０の場合、

　　ｋ＝－１であり、ｎ＝－ｐ＋ｍとなり、ｍ＝ｎ＋ｐ

　ｎ＞０の場合、

　　ｋ＝０であり、ｍ＝ｎ

（－１００００）mod ３２７６８
　　　　＝－１００００＋３２７６８
　　　　＝２２７６８



　ア～エについて、余りを求めると

アの場合、－１００００
イの場合、－２２７６８＋３２７６８＝１００００
ウの場合、１００００
エの場合、２２７６８となる。

　求める答えはエとなる。

◇問26解答　エ

問27問題

正三角形の内部の点から，各辺に下ろした垂線の長さの和は一定である(図１参照）。三角グラフは，この性質を利用して，三つの辺に対応させた要素の割合を各辺への垂線の長さとして表したグラフである。

図２の三角グラフは，３種類のソフトについて，Ａ～Ｄの４人の使用率を図示したものである。正しい解釈はどれか。

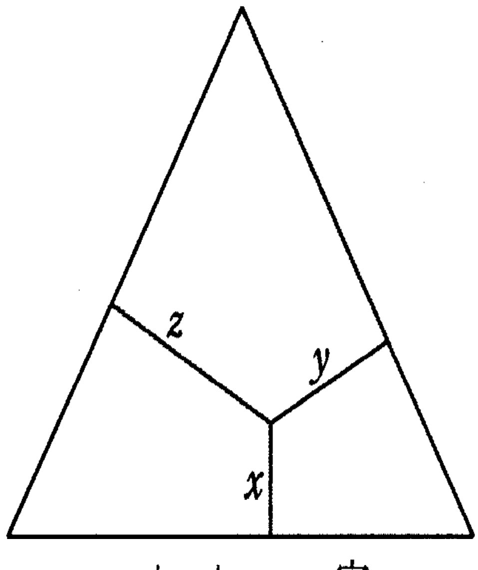


図1　正三角形の性質

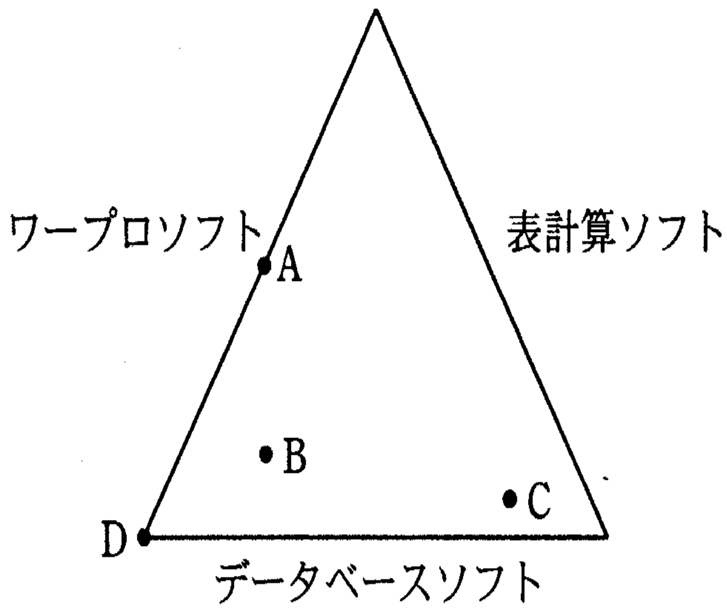


図2　三角グラフ

ア　Ａさんは，ワープロソフトだけを使用している。
イ　Ｂさんは，はかのソフトに比べて表計算ソフトの使用率が高い。
ウ　Ｃさんは，データベースソフト，表計算ソフト，ワープロソフトの順に使用率が高い。
エ　Ｄさんは，表計算ソフトを使用していない。

◇問27解説

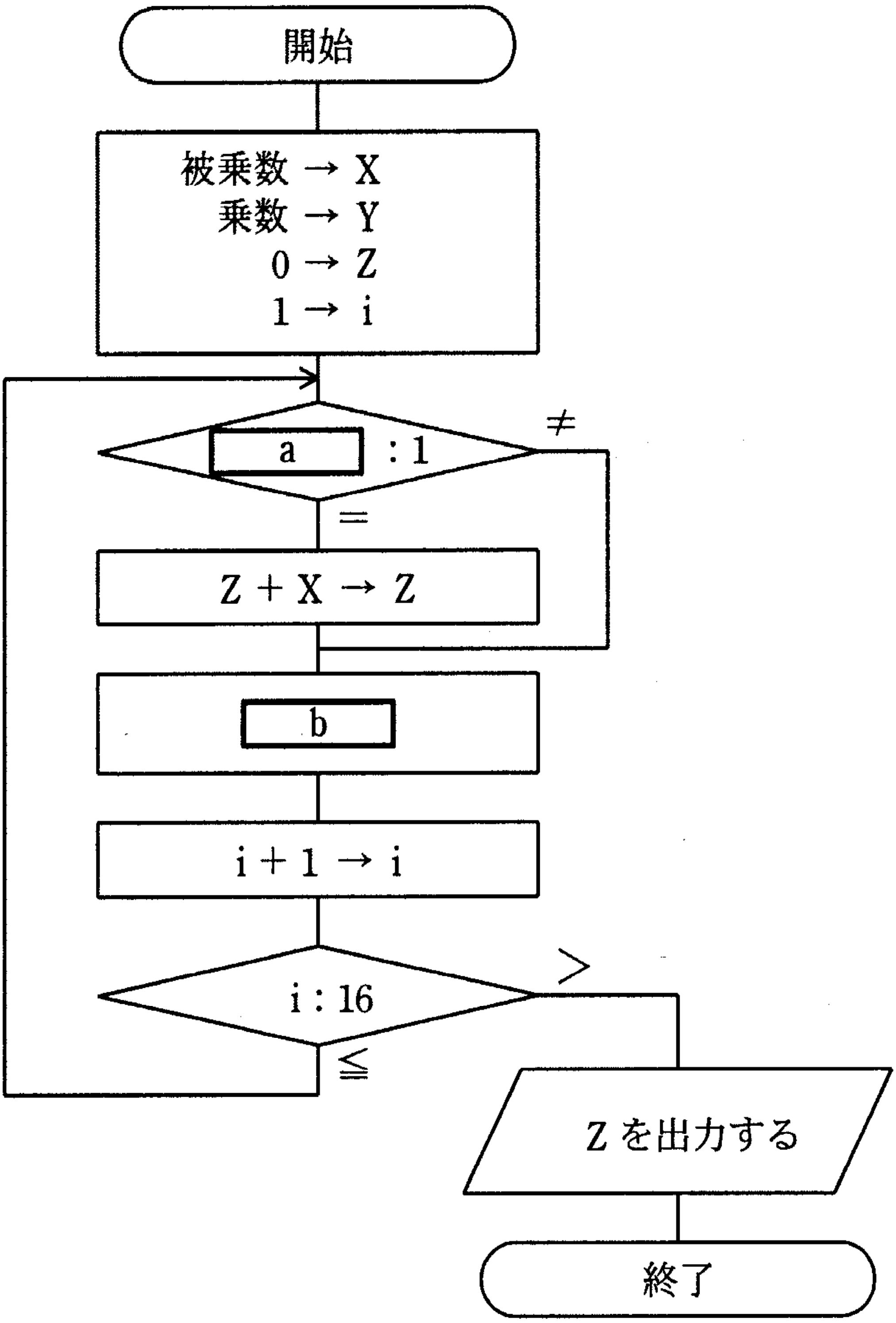
三角グラフの使用法に関する問題である。

Ａ、Ｂ、Ｃ、Ｄの４人のワープロソフト、表計算ソフト、データベースソフトの使用率が問題になっている。使用率はそれぞれの点から関係するソフトの辺に下した垂線の長さで決まる。各辺への垂線の長さの和は一定であるから、これを１００として解釈すればよい。

アのＡはワープロだけ使用しているは、表計算とデータベースを使用し、ワープロは使用していない。
イの表計算ソフトの使用率が高いは、点Ｂから表計算ソフトの辺への垂線が、他の辺への垂線の長さよりも長いため正しい。求める答えはイとなる。
ウのＣの使用率の高い順は、ワープロソフト、表計算ソフト、データベースソフトの順となる。
エの表計算ソフトだけを使用している。

◇問27解答　イ

問28問題



流れ図は，シフト演算と加算の繰返しによって２進数の乗算を行う手順を表したものである。この流れ図中のａ，ｂの処理の組合せとして，正しいものはどれか。

ここで，乗数と被乗数は符号なしの１６ビットで表される。Ｘ，Ｙ，Ｚは３２ビットのレジスタであり，けた送りには論理シフトを用いる。

	a	b
ア	Ｙの最下位ビット	Ｘを１ビット左シフト，Ｙを１ビット右シフト
イ	Ｙの最下位ビット	Ｘを１ビット右シフト，Ｙを１ビット左シフト
ウ	Ｙの最上位ビット	Ｘを１ビット左シフト，Ｙを１ビット右シフト
エ	Ｙの最上位ビット	Ｘを１ビット右シフト，Ｙを１ビット左シフト

◇問28解説

２進数の乗算に関する流れ図の問題である。

２進数の乗算の手順

① 乗数Ｙの最下位ビットが１の時は、乗算結果ＺにＸを加算する。乗数Ｙの最下位ビットが０の時は、何もしない。
② 乗数Ｙを１ビット右にシフトし、被乗数Ｘを左に１ビットシフトする。
③ ｉ＋１→ｉを求める。
④ ｉ≦１６ならば、①に戻り、ｉ＞１６になると、結果を出力して終了する

ａはＹの最下位ビット、ｂはＸを１ビット左シフトして、Ｙを１ビット右シフトする。求める答えはアとなる。

◇問28解答　ア

問29問題

次の関数 f（n，k）がある。f（4， 2）の値は幾らか。

$$
f(n,k) = \begin{cases} 1 & (k=0) \\ f(n-1,k-1)+f(n-1,k) & (0<k<n) \\ 1 & (k=n) \end{cases}
$$

ア　3
イ　4
ウ　5
エ　6

◇問29解説

関数に関する問題である。

$$
\begin{aligned}
f(4,2) &= f(3,1)+f(3,2) \\
&= f(2,0)+f(2,1)+f(2,1)+f(2,2) \\
&= 1+2f(2,1)+1 \\
&= 2+2\times(f(1,0)+f(1,1)) \\
&= 2+2\times 2=6
\end{aligned}
$$

求める答えはエとなる。

◇問29解答　エ

問30問題

整数ｘ，ｙ（ｘ＞ｙ≧０）に対して，次のように定義された関数Ｆ（ｘ，ｙ）がある。

Ｆ（２３１，１５）の値は幾らか。

ここで，ｘmodｙはｘをｙで割った余りである。

$$F(x,y)=\begin{cases} x & (y=0\text{のとき}) \\ F(y,\ x \bmod y) & (y>0\text{のとき}) \end{cases}$$

ア　２
イ　３
ウ　５
エ　７

◇問30解説

関数の問題である。

ｘ＝２３１、ｙ＝１５であるから、ｙ＞０で、

Ｆ（２３１，１５）＝Ｆ（１５，６）
＝Ｆ（６，３）
＝Ｆ（３，０）＝３

求める答はイとなる。

◇問30解答　イ

問31問題

方程式ｆ（ｘ）＝０の解の近似値を求めるアルゴリズムとして知られているニュートン法に関する記述として，適切なものはどれか。

ア　関数ｆ（ｘ）が微分不可能であっても，解の近似値を求めることができる。
イ　幾何学的には，ｙ＝ｆ（ｘ）の接線を利用して解の近似値を求めるものである。
ウ　異なる初期値を二つ与える必要がある。
エ　どのような初期値を与えても，必ず解の近似値が得られる。

◇問31解説

ニュートン法のアルゴリズムに関する問題である。

ニュートン法は、高次方程式の近似解が分かっているとき、近似解を繰り返して修正していき、真の解を求める方式である。

ｙ＝ｆ（ｘ）上の点Ｐ１（ｘ１、ｙ１）の接線を求め、その接線とｘ軸の交点ｘ２を求める。次に、ｙ＝ｆ（ｘ）上の点Ｐ２（ｘ２、ｙ２）の接線を求め、その接線とｘ軸の交点ｘ３を求める。
この操作を繰り返して、隣接する交点の差が決められた値よりも小さくなると、その交点をｆ（ｘ）＝０の解とする考え方である。
従って、ｙ＝ｆ（ｘ）の接線を利用して解の近似値を求める方法である。求める答えはイとなる。

◇問31解答　イ

問32問題

　整列済みの列の末尾から比較して、次の要素の挿入位置を決める単純挿入整列法について考える。
　昇順に整列済みの大きさｎのデータ列を，改めて昇順に整列する処理を行う場合の比較回数のオーダは，どれか。

ア　n
イ　n^2
ウ　$\log n$
エ　$n \log n$

◇問32解説

　整列の挿入法に関する問題である。

　整列済みの要素を最後尾から比較して、再整列する場合の比較回数は、整列対象の要素とその前の要素との比較で挿入位置が決まるため、大きさｎの要素数の比較回数はｎ－１となる。
　従って、比較回数のオーダはｎとなる。求める答えはアとなる。

◇問32解答　ア

問33問題

従業員番号と氏名の対がｎ件格納されている表に線形探索法を用いて、与えられた従業員番号から氏名を検索する。この処理における平均比較回数を求める式はどれか。

ここで、検索する従業員番号はランダムに出現し、探索は常に表の先頭から行う。また、与えられた従業員番号がこの表に存在しない確率をａとする。

ア $\dfrac{n+1}{2}+\dfrac{na}{2}$

イ $\dfrac{(n+1)(1-a)}{2}$

ウ $\dfrac{(n+1)(1-a)}{2}+\dfrac{n}{2}$

エ $\dfrac{(n+1)(1-a)}{2}+na$

◇問33解説

線形探索に関する問題である。

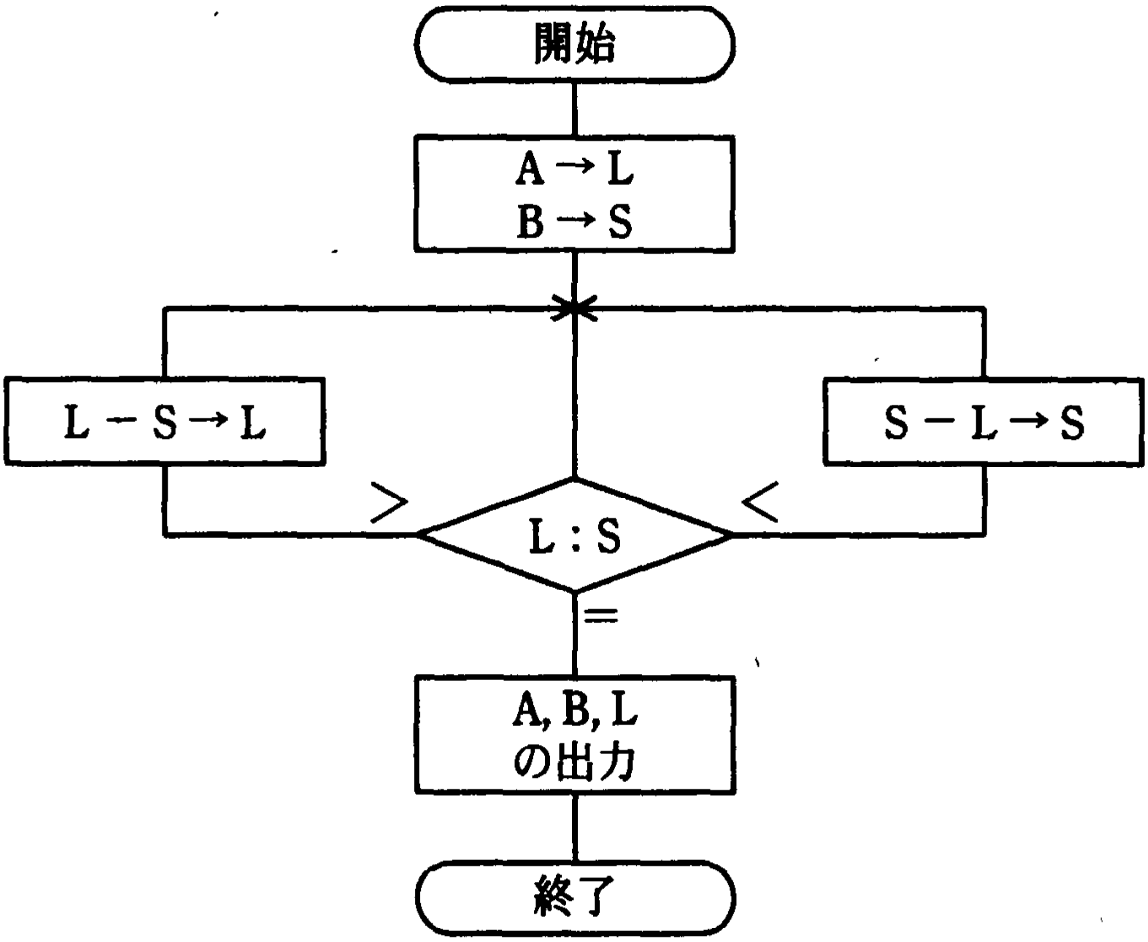
従業員番号が表に存在する確率は１－ａであるから、

① 見つかる場合の平均比較回数は（ｎ＋１）／２であり、発生確率を考えると、（ｎ＋１）（１－ａ）／２となる。
② 見つからない場合の比較回数はｎであり、発生確率を考えるとｎａとなる。

従って、この処理における平均比較回数は
（ｎ＋１）（１－ａ）／２＋ｎａ
となる。求める答えはエとなる。

◇問33解答　エ

問34問題



流れ図は，２数Ａ，Ｂの最大公約数を求めるユークリッドの互除法を，引き算の繰返しによって計算するものである。Ａが８７６、Ｂが２０４のとき，何回の比較で処理は終了するか。

ア　４
イ　９
ウ　１０
エ　１１

◇問34解説

ユークリッド互除法の流れ図に関する問題である。

①　Ｌ＝８７６　Ｓ＝２０４　Ｌ＞Ｓ
　　Ｌ－Ｓ＝８７６－２０４＝６７２→Ｌ
②　Ｌ＝６７２　Ｓ＝２０４　Ｌ＞Ｓ
　　Ｌ－Ｓ＝６７２－２０４＝４６８→Ｌ
③　Ｌ＝４６８　Ｓ＝２０４　Ｌ＞Ｓ
　　Ｌ－Ｓ＝４６８－２０４＝２６４→Ｌ
④　Ｌ＝２６４　Ｓ＝２０４　Ｌ＞Ｓ
　　Ｌ－Ｓ＝２６４－２０４＝６０→Ｌ
⑤　Ｌ＝６０　Ｓ＝２０４　Ｌ＜Ｓ
　　Ｓ－Ｌ＝２０４－６０＝１４４→Ｓ
⑥　Ｌ＝６０　Ｓ＝１４４　Ｌ＜Ｓ
　　Ｓ－Ｌ＝１４４－６０＝８４→Ｓ
⑦　Ｌ＝６０　Ｓ＝８４　Ｌ＜Ｓ
　　Ｓ－Ｌ＝８４－６０＝２４→Ｓ
⑧　Ｌ＝６０　Ｓ＝２４　Ｌ＞Ｓ
　　Ｌ－Ｓ＝６０－２４＝３６→Ｌ
⑨　Ｌ＝３６　Ｓ＝２４　Ｌ＞Ｓ
　　Ｌ－Ｓ＝３６－２４＝１２→Ｌ
⑩　Ｌ＝１２　Ｓ＝２４　Ｌ＜Ｓ
　　Ｓ－Ｌ＝２４－１２＝１２→Ｓ
⑪　Ｌ＝１２　Ｓ＝１２　Ｌ＝Ｓ
　　Ｓ－Ｌ＝１２－１２＝０

従って、１１回となり、求める答えはエとなる。

◇問34解答　エ