【１】流れ図（フローチャート）　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

あらかじめ決められている記号を処理の流れに従って組み合わせることで、アルゴリズムを表現する。流れ図に使われる記号には次のものがある。

|  |  |  |
| --- | --- | --- |
| 記号 | 名称 | 説明 |
|  | 端子 | 流れ図の開始と終了を表す |
|  | 処理 | 実行する処理を表す |
|  | 判断 | 条件によって次に行う処理を選択するという処理を表す |
|  | 線 | 処理の流れを表す |
|  | ループ始端 | 繰り返しの始まりを表す |
|  | ループ終端 | 繰り返しの終わりを表す |

【２】基本制御構造　　　　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

プログラムの基本的な制御構造は、順次、選択、繰返しである。

　［①　　　　］構造

　　各計算や操作が直線的につながっている

　［②　　　　］構造

　　条件により処理内容が分かれる

　［③　　　　］構造

　　終了条件を満たすまで（または繰返し条件を満たしている間）、一連の処理を繰り返す

ループ名

終了条件

(又は継続条件）

処理

ループ名

条件

処理１

処理２

真

偽

処理１

処理２

処理３

順次構造

選択構造

繰返し構造

≪範例≫

　ｘとｙを自然数とするとき，流れ図で表される手続を実行した結果として，適切なものはどれか。

ｒ←ｒ－ｙ

ｑ←ｑ＋１

終了

開始

ｑ←０

ｒ←ｘ

ｒ＜ｙ

Yes

No

|  |  |  |
| --- | --- | --- |
|  | ｑの値 | ｒの値 |
| ア | ｘ÷ｙの余り | ｘ÷ｙの商 |
| イ | ｘ÷ｙの商 | ｘ÷ｙの余り |
| ウ | ｙ÷ｘの余り | ｙ÷ｘの商 |
| エ | ｙ÷ｘの商 | ｙ÷ｘの余り |

≪解答≫　イ

　流れ図では、変数rにxの値が代入され、その変数rからyが繰り返し引かれています。引く都度変数qは1ずつ増加しています。例えば、x＝17、y＝5の場合で考えてみると、1回目：q＝1、r＝12、

２回目：q＝2、r＝7、３回目：q＝3、r＝2となります。よって、変数qはx÷yの商を、変数rはx÷yの余りを表し、解答はイとなります。

【３】ソート（整列）　　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

データ列をある規則に従って並び替えること。なお、小さい順に並べることを［①　　　　］、大きい順に並べることを［②　　　　］という。

１．バブルソート（基本交換法）

データ列の端から順番に隣り合ったデータ同士を比較し、順番が逆なら交換する。この操作を繰り返すことによって整列を行う方法。

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |  | 1 | 2 | 3 | 4 | 5 | 6 |
| 60 | 40  ① | 20 | 50  ④  ③ | 10 | 30  ⑤ |  | 50 | 40 | 10 | 30 | 60 | 20 |
|  | ② |  |  |  |  |  |  |  |  |  |  |
| 40 | 20  ① | 50  ② | 10  ③ | 30 | 60 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 20 | 40  ① | 10  ② | 30 | 50 | 60 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 20 | 10  ① | 30 | 40 | 50 | 60 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | 20 | 30 | 40 | 50 | 60 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | 20 | 30 | 40 | 50 | 60 |  |  |  |  |  |  |  |

※網掛け箇所のデータから順番に完成する

データ数をｎとすると、比較回数はn(ｎ-1)/2回、計算量はｎ2に比例（*O*(n2)）する。最小比較回数はn-1回となる。なお、*O*（オーダ）とはおおよその量を表すときに使う単位であり、nの増加に伴い、計算量がおもに何に従って増加するかを表す。

２．選択ソート

データの中で最小値（または最大値）を探して配列の端から順に格納する動作を繰り返すことによって整列を行う方法。なお、データ数をnとすると、比較回数はn(n-1)/2回、計算量はn2に比例（*O*(n2)）する。

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |  | 1 | 2 | 3 | 4 | 5 | 6 |
| 60 | 40 | 20 | 50 | 10 | 30 |  | 50 | 40 | 10 | 30 | 60 | 20 |
|  | ① |  |  |  |  |  |  |  |  |  |  |  |
| 10 | 40 | 20 | 50 | 60 | 30 |  |  |  |  |  |  |  |
|  | ① |  |  |  |  |  |  |  |  |  |  |
| 10 | 20 | 40 | 50  ① | 60 | 30 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | 20 | 30 | 50  ① | 60 | 40 |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | 20 | 30 | 40 | 60 | 50  ① |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | 20 | 30 | 40 | 50 | 60 |  |  |  |  |  |  |  |

※網掛け箇所のデータから順番に完成する

３．挿入ソート

データを部分的に整列していき、整列されている列の中に未整列のデータを追加することで整列を行う方法。そのため、最初のデータの並び方によって比較回数が異なる。データ数をnとすると、平均比較回数は約n(n-1)/4回となり、計算量は*O*(ｎ2)になる。

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 |  | 1 | 2 | 3 | 4 | 5 | 6 |
| 60  ① | 40 | 20 | 50 | 10 | 30 |  | 50 | 40 | 10 | 30 | 60 | 20 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 40 | 60 | 20 | 50 | 10 | 30 |  |  |  |  |  |  |  |
|  | ② | ① |  |  |  |  |  |  |  |  |  |  |
| 20 | 40 | 60 | 50 | 10 | 30 |  |  |  |  |  |  |  |
|  |  |  | ① |  |  |  |  |  |  |  |  |  |
| 20 | 40 | 50 | 60 | 10 | 30 |  |  |  |  |  |  |  |
|  | ④ | ③ | ② | ① |  |  |  |  |  |  |  |  |
| 10 | 20 | 40 | 50 | 60 | 30 |  |  |  |  |  |  |  |
|  |  |  | ③ | ② | ① |  |  |  |  |  |  |  |
| 10 | 20 | 30 | 40 | 50 | 60 |  |  |  |  |  |  |  |

※網掛け箇所のデータを網掛け箇所より左側のデータに挿入する

４．その他のソート

|  |  |
| --- | --- |
| 整列名 | 特徴 |
| **クイックソート** | 整列の対象となるデータを、**ある値より小さいデータからなるグループと、ある値より大きいデータからなるグループに分割**し、この操作を各グループともデータ数が１になるまで繰り返すことにより整列を行う方法。 |
| **ヒープソート** | ヒープと呼ばれる**木を用いた整列法**。整列の対象となる要素からヒープを構成し、ヒープの親子関係を正していくことによって整列を行う。**すべての節が必ず親＞子（昇順の場合）または子＞親（降順の場合）となる構造を作る**ことで根がすべての節点の中で最大値（昇順の場合）または最小値（降順の場合）となる。 |
| **シェルソート** | 挿入ソートを改良した方法。**一定間隔（通常は要素数の1/2）離れた要素同士で挿入ソートを行い**、その後間隔を半分にしながら挿入ソートを繰り返す。 |
| **マージソート** | 要素全体を分割し、**各部分内で整列しておき、その後分割した部分を併合（マージ）する**ことで整列する方法。 |

≪範例≫

　整列アルゴリズムの一つであるクイックソートの記述として、適切なものはどれか。

ア　対象集合から基準となる要素を選び、これよりも大きい要素の集合と小さい要素の集合に

　　分割する。この操作を繰り返すことで、整列を行う。

イ　対象集合から最も小さい要素を順次取り出して、整列を行う。

ウ　対象集合から要素を順次取り出し、それまでに取り出した要素の集合に順序関係を保つよ

　うに挿入して、整列を行う。

エ　隣り合う要素を比較し、逆順であれば交換して、整列を行う。

≪解答≫　ア

　イ　選択ソートの記述です。

　ウ　挿入ソートの記述です。

　エ　バブルソート（基本交換法）の記述です。

【４】探索（サーチ）　　　　　　　　　　　　　　　　　　　　　　　　　…テキストP.（　　　）

配列やファイルなどのデータの集まりから、特定のデータを見つけるために走査する過程のことを探索という。

１．線形探索（シーケンシャルサーチ）

特定のデータを探索するために、配列の［①　　　　　　　　］から順番に調べていく方法。配列要素がどのような並び順でも、探索することができる。

89

23

46

77

51

94

35

１

２

３

４

５

６

７

Ｔ

94

先頭から順番に比較

Ｘ



ｎ＝７（配列の大きさ）



例えば、タンスの引き出しから探し物を探すときに、端の引き出しから順番にしらみつぶしに探すことだよ！

線形探索のイメージ

開始

終了

Xにデータを入力

１→ｉ

ｉ＞ｎまたは

Ｘ＝Ｔ[ｉ]

ｉ+１→ｉ

ｉ＞ｎ

“Found”を表示

“Not found”

を表示

Yes

No

・・・Ｘに探したいデータを入力する

・・・変数ｉに探索開始位置（添字）の設定

・・・繰り返しの終了条件は、引き出しをすべて

見終わった、または、探し物が見つかったか

・・・終了条件に当てはまらない間、変数ｉを+１

　　　して次の引き出しの場所を設定する

・・・上の繰り返しで変数ｉの値が配列の大きさｎを超えたのであれば、配列には探し物はなかったことになる。上の繰り返しで変数ｉがｎ以下で止まっているということは探し物が見つかったことになる。

　　　　例　線形探索のフローチャート

２．２分探索（バイナリサーチ）

探索の対象となる配列が、あらかじめ［②　　　　　　　　］に並んでいるときに用いられる。常に配列の探索範囲の中央のデータと比較し、比較したデータよりも前半部分または後半部分を切り捨て、探索する範囲を徐々に狭めていく方法。

①　探索範囲の上限，下限を示す変数をそれぞれＬ，Ｈとし，中央の値Ｍを求める。

最初はＬ＝１，Ｈ＝７，Ｍ＝(Ｌ＋Ｈ)÷２＝(１＋７)÷２＝４（端数が出る場合は，小数点以下切り捨て）である。

②　配列Ｔ(４)とＸを比較する。51＜77なので，Ｌを（Ｌ＝Ｍ＋１）移動して，前半部分を切り捨てる。

③　残った範囲で，ＬとＨからＭを求める。

④　配列Ｔ(６)とＸを比較する。89＞77なので，Ｈを（Ｈ＝Ｍ－１）移動して，後半部分を切り捨てる。

⑤　残った範囲でＬとＨからＭを求める。

⑥　以下，Ｔ(Ｍ)＝Ｘか，Ｌ＞Ｈになるまで，④と⑤を繰り返す。

１

２

３

４

５

６

７

Ｔ

89

23

46

77

51

94

35

77

Ｘ

Ｌ

Ｈ

Ｍ

探索範囲

23

46

51

35

探索範囲

１

２

３

４

５

６

７

Ｔ

89

77

94

77

Ｘ

Ｌ

Ｈ

Ｍ

切捨て

探索範囲

１

２

３

４

５

６

７

Ｔ

89

23

46

77

51

94

35

77

Ｘ

ＬＭＨ

切捨て



例えば、辞書で単語を探す想像をしてみよう！

辞書はアルファベット順やあいうえお順になっているね。最初に辞書の真ん中を開いて、見つけたい単語がそのページより前半か後半か判断して探すイメージだよ！２分探索法は、１回の探索で探索範囲を半分にすることができるんだ。従って、配列のデータが２倍になっても探索回数は１回しか増えないよ。

２分探索のイメージ

＜参考＞　２分探索の流れ図

開始

終了

１→Ｌ

ｎ→Ｈ

Ｔ[Ｍ]：Ｘ

“Found”を表示

“Not found”

を表示

＞

＜

（Ｌ+Ｈ）/2→Ｍ

Ｍ+１→Ｌ

Ｍ-１→Ｈ

Ｔ[Ｍ]：Ｘ

＝

＝

≠

T[M]=X

または

L＞H

Xにデータを入力

・・・Ｘに探したいデータを入力する

・・・探索範囲の下限を変数Ｌに設定

・・・探索範囲の上限を変数Ｈに設定

・・・探索範囲の中央を計算により変数Ｍに設定

・・・探索範囲の中央のデータとデータＸと比較

・・・T[M]＜Xならば中央のデータより探したいデータは後半にある。すなわち、下限の再設定。T[M]＞Xならば中央のデータより探したいデータは前半にある。すなわち、上限の再設定。

繰り返しの条件は見つかったか、見つからなかったか。見つからない条件は、L>H。

例　２分探索のフローチャート（昇順の場合）

|  |  |  |
| --- | --- | --- |
|  | 平均探索回数 | 最大探索回数 |
| 線形探索 | N÷2回 | N回 |
| ２分探索 | log2N回 | log2N+1回 |

探索回数のまとめ（配列の大きさをNとした場合）

※対数log（logarithm）とは？

　　通常、logabと表現され「aを何乗したらbになるか」を求めます。

　　以下の通り、表現方法が変わるだけであるため、難しく考える必要はありません。

52＝25（5を2回かけると？＝25）　＝＝＞　log525＝2（5を何回かけると25になるか？＝2）

　　つまり、２分探索で配列の大きさが16であれば、平均探索回数はlog216＝4回になります。

３．ハッシュ法

データやキーの値から直接、格納位置（添字）を計算する方法であり、データの探索を高速に行うことができる。この方法では、与えられたキーに特別な関数を用いて、格納位置を求める。

このときに用いる関数を［③　　　　　　］といい、ハッシュ関数が返す値をハッシュ値という。また、ハッシュ関数を用いてデータの格納位置を求めることをハッシングという。なお、キーの分布によらず、できるだけ均等でかつ広範囲なハッシュ値を与えるハッシュ関数が良いハッシュ関数といえる。

ハッシュ関数

キー

データ

キー値

ハッシュ値

ハッシュ値で

直接アクセス

ハッシュ表

キー

データ

ハッシュ値が示す位置

ハッシュ法の概念

ハッシュ法は、データを格納する際、ハッシュ関数によって特定の範囲の値にキーを変換するため、異なるキーにもかかわらず同一のハッシュ値が得られることがある。これを［④　　　　　　　　　　］といい、先に格納されていたデータをホームレコード、後から変換されたデータをシノニムレコードという。

ハッシュ法による探索に要する平均比較回数はシノニムレコードが発生する確率に左右される。衝突がまったくない場合には１回の比較で探索できるが、衝突が多くなるとそれだけ比較回数も増加する。

シノニムレコードが発生する確率を最小にするには、ハッシュ値に偏りがなく、一様に分布している必要がある。

次の手順でハッシュ表にデータを格納する場合

ハッシュ関数：ハッシュ値＝（キー値÷10）の余り

ハッシュ表

０

１

２

３

21

12

⋮

①キー値＝12の場合

12÷10＝１余り２

ハッシュ表の２番目に格納→ホームレコード

②キー値＝21の場合

21÷10＝２余り１

ハッシュ表の１番目に格納→ホームレコード

③キー値＝42の場合

42÷10＝４余り２

ハッシュ表の２番目に格納→シノニムレコード

衝突

衝突（例）

≪範例１≫

　顧客番号をキーとして顧客データを検索する場合、２分探索を使用するのが適しているものはどれか。

　ア　顧客番号から求めたハッシュ値が指し示す位置に配置されているデータ構造

　イ　顧客番号に関係なく、ランダムに配置されているデータ構造

　ウ　顧客番号の昇順に配置されているデータ構造

　エ　顧客番号をセルに格納し、セルのアドレス順に配置されているデータ構造

≪解答≫　ウ

　２分探索を行う場合、データはあらかじめ整列している必要があります。

≪範例２≫

表検索におけるハッシュ法の特徴はどれか。

　ア　２分木を用いる方法の一種である。

　イ　格納場所の衝突が発生しない方法である。

　ウ　キーの関数値によって格納場所を決める。

　エ　探索に要する時間は表全体の大きさにほぼ比例する。

≪解答≫　ウ

　ア　２分探索の説明です。

　イ　別のキー値から同じ格納場所が計算され、衝突が起こることがあります。

　エ　ハッシュ法は表の大きさにかかわらず常に一定です。

≪範例３≫

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ｎ2であるとは、ｎ個のデータを処理する時間がｃｎ2（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 | log2n | n | 1 |
| イ | nlog2n | n | log2n |
| ウ | nlog2n | n2 | 1 |
| エ | n2 | 1 | n |

≪解答≫　ア