

問3 マージソートに関する次の記述を読んで、設問1～3に答えよ。

マージソートは、整列（ソート）したいデータ（要素）列を、細かく分割した後、併合（マージ）を繰り返して全体を整列する方法である。

ここでは、それぞれの要素数が1になるまでデータ列の分割を繰り返し、分割されたデータ列を昇順に並ぶように併合していくアルゴリズムを考える。例として、要素数が8の場合のアルゴリズムの流れを図1に示す。

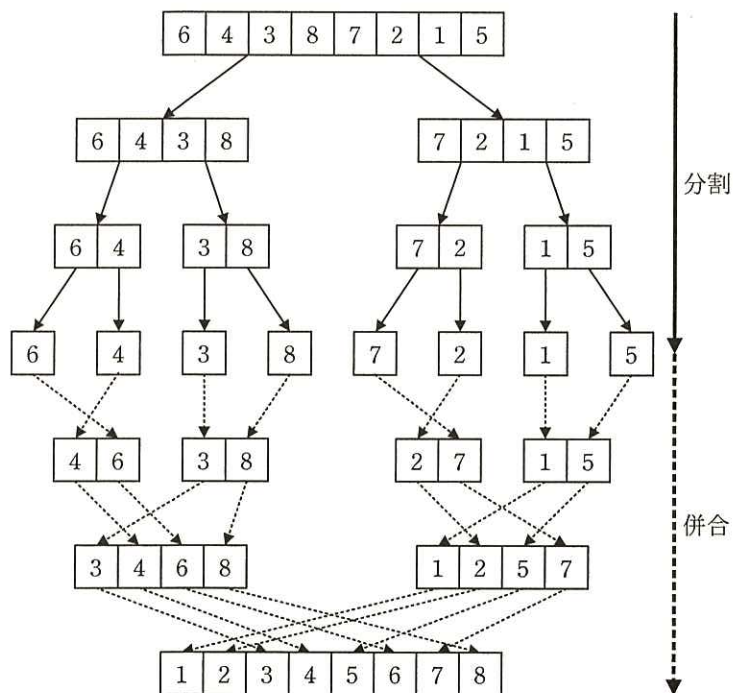


図1 アルゴリズムの流れ

再帰呼び出しを使って記述したマージソートのアルゴリズムを図2に示す。

- (1) 与えられたデータ列の要素数が1以下であれば、整列済みのデータ列とし、呼び出し元に処理を戻す。要素数が2以上であれば、(2)に続く。
- (2) データ列を、要素数がほぼ同じになるよう前半と後半のデータ列に分割する。
- (3) 前半と後半のデータ列に対し、それぞれマージソートのアルゴリズムを再帰的に呼び出す。
- (4) 前半と後半の二つのマージソート済みデータ列を、要素が昇順に並ぶよう一つのデータ列に併合する。

図2 マージソートのアルゴリズム

図2のアルゴリズムを連結リストに対して実行するプログラムを考える。ここでは、整列対象のデータとして正の整数を考える。連結リストは、複数のセルによって構成される。セルは、正の整数値を示すメンバ `value` と、次のセルへのポインタを示すメンバ `next` によって構成される。連結リストの最後のセルの `next` の値は、NULLである。連結リストのデータ構造を図3に示す。

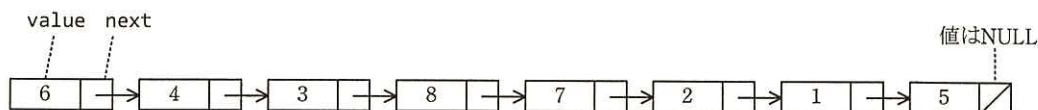


図3 連結リストのデータ構造

[連結リストの分割]

図2中の(2)の処理を行う関数 `divide` を考える。関数 `divide` は、連結リストの先頭へのポインタ変数 `list` を引数とし、分割後の後半の連結リストの先頭へのポインタを戻り値とする。連結リストの分割前後のイメージを図4に示す。

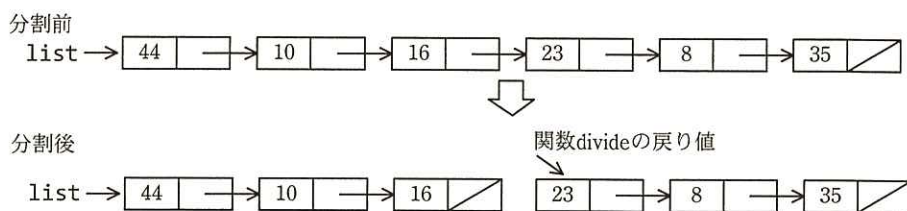


図4 連結リストの分割前後のイメージ

連結リストをセルの個数がほぼ同じになるように分割するために、ポインタ変数を二つ用意し、一方が一つ進むごとに、他方を二つずつ進める。後者のポインタが連結リストの終わりに達するまでこの処理を繰り返すと、前者のポインタは連結リストのほぼ中央のセルを指す。この方法を利用した関数 `divide` のプログラムを図5に示す。

以下、連結リストのセルを指すポインタ変数を `a` とするとき、`a` が指すセルのメンバ `value` を `a->value` と表記する。

```

function divide( list )
  a ← list // a はセルへのポインタ
  b ← a->next // b はセルへのポインタ
  if ( b が NULL と等しくない )
    b ← b->next
  endif

  while (  ) // 連結リストの終わりまで繰り返す
    a ← a->next
    b ← b->next
    if ( b が NULL と等しくない )
      
    endif
  endwhile

  p ← a->next // p はセルへのポインタ
   ← NULL
  return p
endfunction

```

図 5 関数 divide のプログラム

[連結リストの併合]

図 2 中の(4)の処理を行う関数 merge を考える。関数 merge は、二つの連結リストの先頭へのポインタ変数 a と b を引数とし、併合後の連結リストの先頭へのポインタを戻り値とする。併合処理を行う際には、ダミーのセルを用意し（そのセルへのポインタを head とする）、この後ろに併合後の連結リストを構成する。a と b が指すセルの値を比較しながら、値が小さい順に並ぶよう処理を進める。連結リストの併合の流れを図 6（処理は、①、②、③、…と続く）に、関数 merge のプログラムを図 7 に示す。

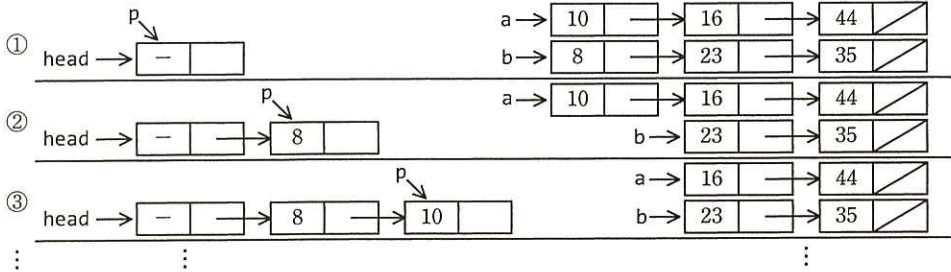


図 6 連結リストの併合の流れ

```

function merge( a, b )      // a, b は併合対象の連結リストの先頭へのポインタ
    ダミーのセルを用意する
    head ← ダミーのセルへのポインタ
    p ← head

    while (  , かつ, b が NULL と等しくない )
        if ( a->value が b->value 以下である )
            p->next ← a
            p ← a
            a ← a->next
        else
            p->next ← b
            p ← b
            b ← b->next
        endif
    endwhile

    if (  )      // 要素が残っている連結リストを連結する
        p->next ← b
    else
        p->next ← a
    endif

    return 
endfunction

```

図7 関数 merge のプログラム

設問1 [連結リストの分割] について, (1)~(3)に答えよ。

- (1) 図5中の ~ に入れる適切な字句を答えよ。
- (2) 図3の連結リストに対して関数 divide を実行し, プログラムが図5中の α の部分に達したとき, ポインタ変数 a は, 図3中のどのセルを指しているか。指しているセルの値 (value の数値) を答えよ。
- (3) 奇数 $2N+1$ 個のセルから成る連結リストを関数 divide で分割すると, 前半と後半の連結リストのセルの個数はそれぞれ幾つになるか式で答えよ。

設問2 図7中の ~ に入れる適切な字句を答えよ。

設問3 32 個のセルから成る連結リストに対し, 図2 のアルゴリズムに相当するプログラムを実行した場合, 関数 merge は何回呼び出されるか答えよ。