示現塾　プロジェクトマネージャ・テクニカルエンジニア（ネットワーク）など各種セミナーを開催中！！
開催日、受講料、カリキュラム等、詳しくは、http://zigen.cosmoconsulting.co.jp　今すぐアクセス！！

平成１９年度　秋期　ソフトウェア開発技術者　午後Ⅱ問題

問１　B 木に関する次の記述を読んで，設問 1～4 に答えよ。

B 木は，外部記憶装置にデータを格納するために考えられた，木構造のデータ構造である。ここでは，データを識別するキーを正の整数として，キーだけを B 木に格納する場合を考える。ただし，キーの重複はないものとする。

B 木の各ノードは図 1 に示す構造をしており，木全体として次の(1)～(8)の性質をもっている。ただし，一つのノードにはキーを n 個まで格納できるものとする。また，n は偶数で，M＝n／2 とする。

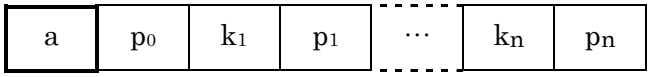


図１　ノードの構造

(1) k_1～k_n にはキーが格納される。また，ノード内に格納されているキーの個数が a に格納される。
(2) ノード内では，キーは昇順に並んでいる。つまり，$k_1 < k_2 < \cdots < k_a$ である。
(3) p_0 ～p_n には子ノードへのポインタが格納される。
(4) p_0 が指す部分木内のキーはすべて k_1 より小さく，$1 \leqq i < a$ の p_i が指す部分木内のキーは，すべて k_i より大きくかつ k_{i+1} より小さい。また，p_a が指す部分木内のキーは，すべて k_a より大きい。
(5) 子ノードが存在しない場合，該当するポインタは nil である。
(6) 根のノードには 1～n 個のキーが，根以外のノードには M～n 個のキーが格納されている。つまり，根以外のノードでは，キーは格納できる個数の半数以上格納されている。
(7) 葉のノード以外では，ノード内のキーの個数に 1 を加えた個数のポインタが格納されている。
(8) 根から葉のノードまでの経路の長さ（深さ）は，どの葉のノードも同じである。

n=2（M=1）の場合の B 木の例を図 2 に示す。

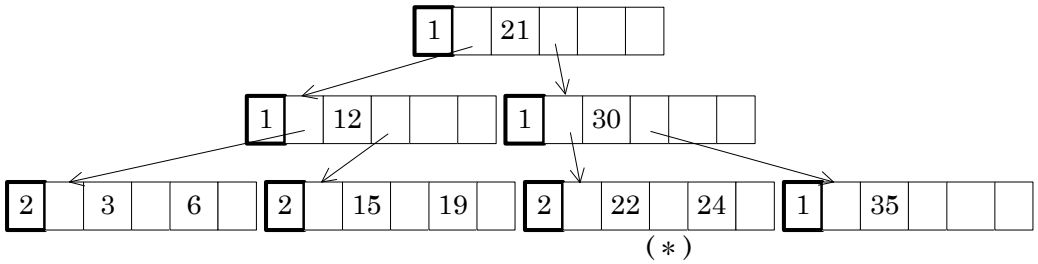


図２　Ｂ木の例

B 木に対する操作として，指定されたキーが B 木内にあるかどうかを調べる“探索”と，指定されたキーを B 木に格納する“挿入”の二つの操作を考える。

示現塾　プロジェクトマネージャ・テクニカルエンジニア（ネットワーク）など各種セミナーを開催中！！
開催日、受講料、カリキュラム等、詳しくは、http://zigen.cosmoconsulting.co.jp　今すぐアクセス！！

〔探索〕

指定されたキーx について，根のノードから順に，各ノードに格納されているキーk_i と x を順番に比較しながら，次の動作を行う。

・$x = k_i$ となる k_i が見つかれば目的の値が見つかり探索終了
・$x < k_1$ ならば，ポインタ p_0 をたどって，次のノードを探索
・$k_i < x < k_{i+1}$（$1 \leqq i < a$）ならば，ポインタ p_i をたどって，次のノードを探索
・$k_a < x$ ならば，ポインタ p_a をたどって，次のノードを探索
・たどろうとしたポインタが nil ならば，目的の値は見つからずに探索終了

〔挿入〕

指定されたキーx を挿入する場合，上の〔探索〕と同様にして，キーx が入る可能性があるノードを探索する。探索しているノードでキーx が見つかったときは，何もせずに終了する。一方，葉のノードに到達してもキーx が見つからないときは，到達した葉のノード，又は新たに用意するノードに x を挿入する。すなわち x をノードに挿入する場合，次の二つの可能性が考えられる。

（1）到達したノードに新たにキーを挿入する場所がある場合，そのノードの適切な場所に x を挿入する。

（2）到達したノードに新たにキーを挿入する場所がない場合，新たにノードを用意し，到達したノードに格納されているキーと x の合計 n＋1（＝2M＋1）個のキーを昇順に並べて再配置する（分割）。再配置は，前から M 個のキーと後ろから M 個のキーに分けてそれぞれを別のノードに格納し，中央の一つのキーを親のノードの適切な場所に格納する（昇進）。親のノードに新たにキーを挿入する場所がない場合は，親のノードに対して分割と昇進を繰り返す。昇進させるための親のノードがない場合は，親のノードを新たに用意してキーを適切な場所に格納する（新しい根になる）。

（1），（2）の処理の終了後，処理の対象となった各ノードに格納されているキーの個数を更新する。

分割と昇進の例として，図 2 の B 木に 26 を挿入する場合を説明する。

図 2 内で 26 を挿入すべきノードを根から探索していくと，図 2 で（*）を付けたキー22 と 24 が既に格納されているノードが挿入対象となる。しかし，このノードには既に 2M（＝n）個のキーが格納されており，新たにキーを挿入する場所はない（図 3（a））。この場合，挿入対象のノード内にある 2M 個のキーに，新たに挿入したいキーを加えた全部で 2M＋1 個のキーを昇順に並べ，並べたキーを前半の M＋1 個（22 と 24）と後半の M 個（26）の二つに分ける。そして，元々の挿入対象のノードに前半の M＋1 個のキーとポインタを格納する。さらに新しくノードを一つ作成し，そのノードに後半の M 個のキーとポインタを格納する（図 3（b））。ただし，この例では分割されたノードは葉のノードであるので，ポインタには nil が入る。そして，M＋1 個のキーが格納されているノードの中で最も大きなキー（24）の値を新しいノードへのポインタとともに親のノードに渡し，渡されたキーとポインタを親のノード内の適切な場所に格納する（図 3（c））。

示現塾　プロジェクトマネージャ・テクニカルエンジニア（ネットワーク）など各種セミナーを開催中！！
開催日、受講料、カリキュラム等、詳しくは、http://zigen.cosmoconsulting.co.jp　今すぐアクセス！！

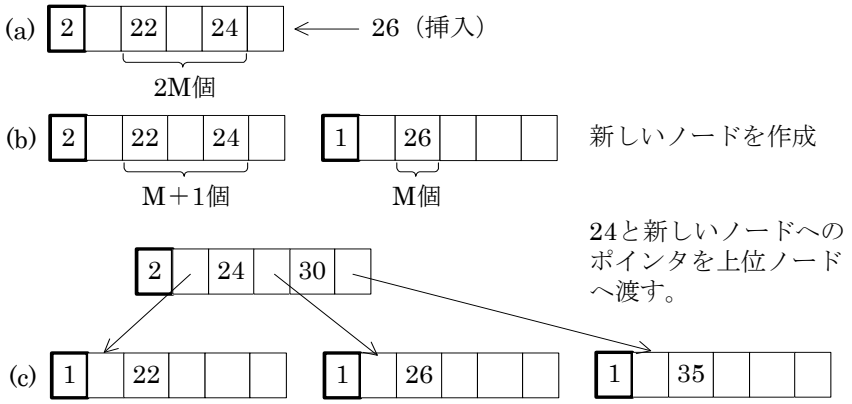


図３　分割と昇進

B 木のノードをプログラムで扱うために，データ構造 node を定義する。node は図 4 に示すように，node 内に保持しているキーの個数をもつ変数 n_key と，それぞれ 2M，2M＋1 個の配列変数である key，branch から構成される。key にはノード内のキーが，branch にはポインタが入る。初期値として，branch には nil を入れる。p をデータ構造 node を指すポインタとして，node 内の変数は，それぞれ p->n_key，p->key[i](1≦i≦2M)，p->branch[j](0≦j≦2M) と表記する。

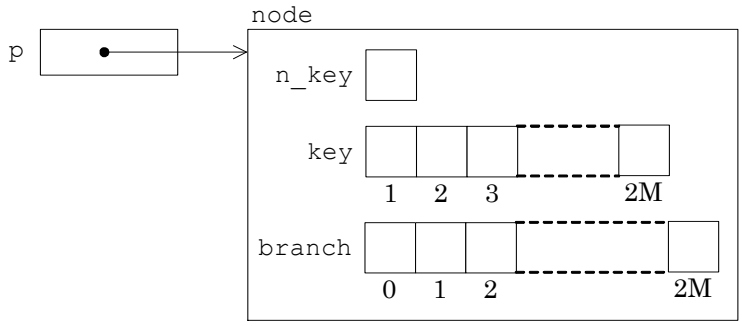


図４　データ構造 node とポインタ p

まず，探索の操作を行うプログラムを考える。図 5 は探索を行う関数 search のプログラムで，探索する B 木の根のノードへのポインタ p と探索するキーの値 key value を受け取り，B 木内にキーが見つかった場合は TRUE を，見つからなかった場合は FALSE を返す。

示現塾　プロジェクトマネージャ・テクニカルエンジニア（ネットワーク）など各種セミナーを開催中！！
開催日、受講料、カリキュラム等、詳しくは、http://zigen.cosmoconsulting.co.jp　今すぐアクセス！！

```
function search( p, keyvalue )
//p: 根のノードへのポインタ，keyvalue: 探索するキーの値
  while( p が [ ア ] )
    k ← 1
    while( k が P->n_key 以下である かつ [ イ ] )
      k ← k+1
    endwhile
    if( k が p->n_key 以下である かつ p->key[k]が keyvalue に等しい )
      return TRUE                    //キーが見つかった
    endif
    p ← [ ウ ]
  endwhile
  return FALSE                  //キーが見つからない
endfunction
```

図５　関数 search のプログラム

次に，挿入の操作を行うプログラムを考える。ただし，ここではプログラム全体ではなく，〔挿入〕(1)の場合に使用する関数 insertitem と，〔挿入〕(2)の場合に使用する関数 split について考えるものとする。

図 6 に関数 insertitem のプログラムを示す。関数 insertitem は，対象ノードに新たにキーとポインタを格納する場所がある場合に，挿入したいキーとポインタをノードの k 番目の要素として格納する。したがって，ノードの k 番目以降のキーとポインタを一つずつ後ろにずらし，引数で渡される挿入するキーの値 keyvalue とポインタの値 newp をノードの k 番目の要素に代入する。

```
function insertitem( P, k, keyvalue, newp )
// p: ノードへのポインタ，k: 挿入場所の配列の添字
// keyvalue: 挿入するキーの値，newp: 挿入するポインタの値
  for( i を p->n_key から k まで 1 ずつ減らす )
    P->key[ [ エ ] ] ← P->key[ [ オ ] ]
    P->branch[ [ エ ] ] ← P->branch[ [ オ ] ]
  endfor
  P->key[k] ← keyvalue
  P->branch[k] ← newp
  P->n_key ← [ カ ]
endfunction
```

図６　関数 insertitem のプログラム

図 7 に関数 split のプログラムを示す。関数 split は，ポインタ p が指すノードに新しくキーを挿入したいが，対象ノードには既に 2M 個のキーが格納されており，キーを新たに格納する場所がない場

示現塾　プロジェクトマネージャ・テクニカルエンジニア（ネットワーク）など各種セミナーを開催中！！
開催日、受講料、カリキュラム等、詳しくは、http://zigen.cosmoconsulting.co.jp　今すぐアクセス！！

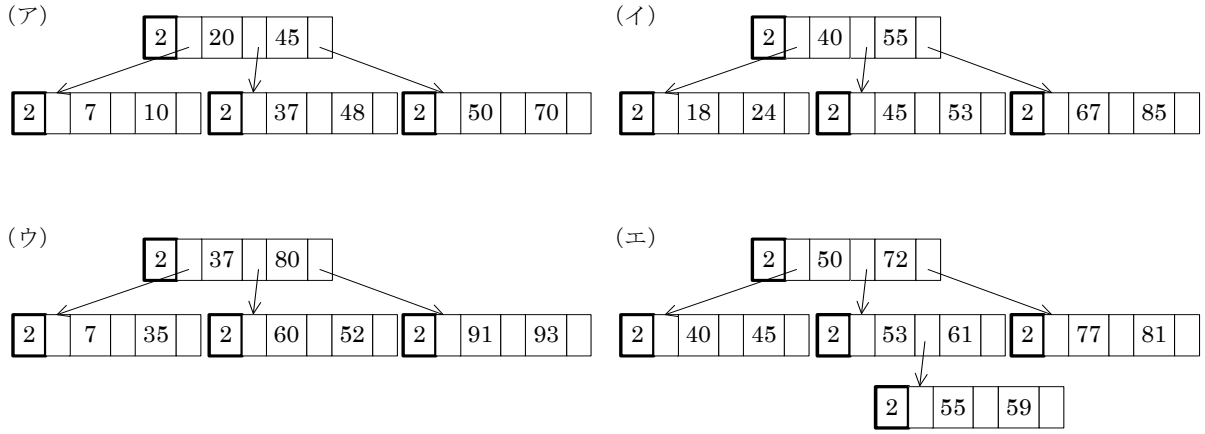
合に，分割を行う関数である。このとき，元々の対象ノードに入っていたキーと新たに挿入するキーを合わせた全部で 2M＋1 個のキーを昇順で並べたとき，新たに挿入するキーは k 番目のキーとなっている状況で呼び出される。関数 split は，2M＋1 個のキーを，M＋1 個と M 個の二つの組に分け，前半の M＋1 個のキーをポインタ p が指すノードに格納し，後半の M 個のキーを新しく生成するポインタ q が指すノードに格納する。また，ノード内のポインタもそれに合わせて適切に格納する。そして，親ノードへ昇進するキーの値を大域変数の parentkeyvalue に設定し，新たに作成したノードへのポインタ q を返り値とする。

```
function split( P, k, keyvalue, newp )
// p: ノードへのポインタ，k: 昇順でキーを並べたときの挿入するキーの順番
// keyvalue:挿入するキーの値，newp:挿入するポインタの値
  if( k が M＋1 以下である )
    m ← M
  else
    m ← M＋1
  endif
  新しくノードを作成し，作成したノードへのポインタを q に代入
  for( j を m＋1 から 2*M まで 1 ずつ増やす )       // 後半部の移動
    q->key[ キ ] ← P->key[j]
    q->branch[ キ ] ← p->branch[j]
  endfor
  q->n_key ← ク
  P->n_key ← ケ
  if( k が M＋1 以下である )
    insertitem( p, k, keyvalue, newp )
  else
    insertitem( q, k-m, keyvalue, newp )
  endif
  parentkeyvalue ← コ                    // 昇進するキーの値を設定
  q->branch[0] ← p->branch[p->n_key]
  P->n_key ← サ
  return q                    // 新しく作成したノードへのポインタを返す
endfunction
```

図７　関数 split のプログラム

示現塾　プロジェクトマネージャ・テクニカルエンジニア（ネットワーク）など各種セミナーを開催中！！
開催日、受講料、カリキュラム等、詳しくは、http://zigen.cosmoconsulting.co.jp　今すぐアクセス！！

設問１　次の（ア）～（エ）の図の中でＢ木として正しいものはどれか，記号で答えよ。



設問２　M=1 である図８の B 木に 35, 40 を順に挿入した後の B 木はどうなるかを示せ。必要ならば，ノードを追加せよ。

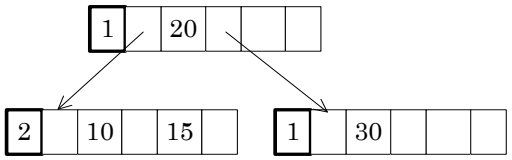


図８　挿入前のＢ木

設問３　本文中のプログラムに関して，(1)，(2) に答えよ。

(1) 図 5 のプログラム中の［　ア　］～［　ウ　］に入れる適切な字句を答えよ。

(2) 図 6，7 のプログラム中の［　エ　］～［　サ　］に入れる適切な字句を答えよ。

示現塾　プロジェクトマネージャ・テクニカルエンジニア（ネットワーク）など各種セミナーを開催中！！
開催日、受講料、カリキュラム等、詳しくは、http://zigen.cosmoconsulting.co.jp　今すぐアクセス！！

設問４　木全体のキーの個数が最小となる場合は，根に１個，根以外のすべてのノードにキーがそれぞれ M 個入っている場合である。この場合の B 木中のキーの個数の評価に関して，(1) ～ (3) に答えよ。

(1) 図 9 に示すように，同じ深さごとにノード数とキーの数を考える。図 9 中の［　シ　］～［　ソ　］に入れる適切な式を答えよ。

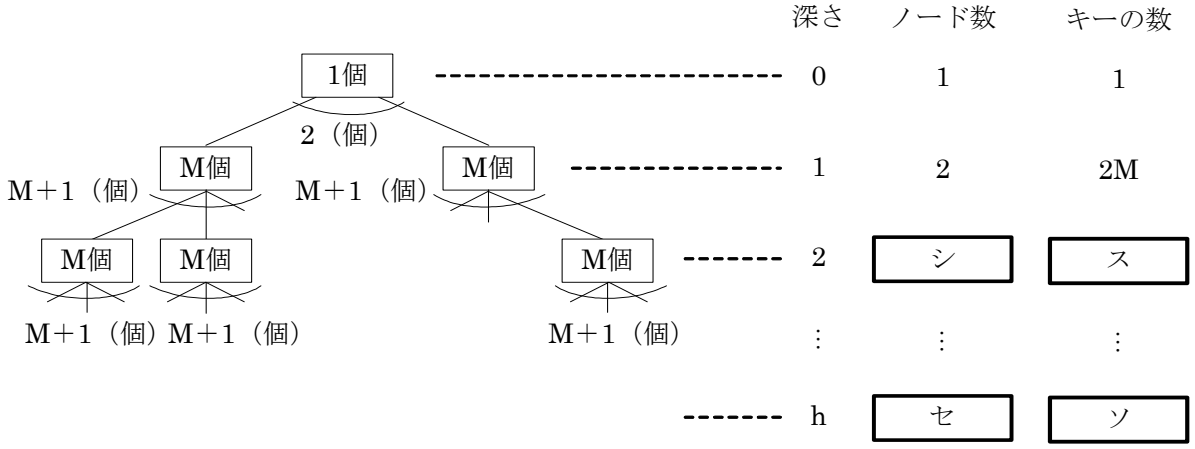


図９　ノードの数とキーの数

(2) 深さ h までのキーの総数を答えよ。

(3) 全部のキーの個数を 100 万個，M を 255 とすると B 木の深さは，最大で幾らになるか答えよ。