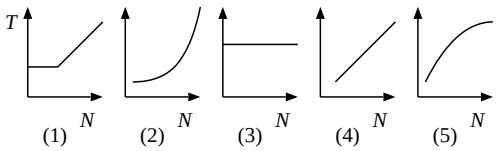
情報構造論 (斉藤)　後期期末試験　　番号：
4EI　2009/02/06　(1/3)　　氏名：

6 つの設問の中から 5 題を選んで解答せよ。

1　一般総括問題 (5 × 4)

データ件数 N の増加と共に、処理時間の変化のグラフの概形を右図にいくつか示した。以下の設問に示す処理の時間変化のグラフにふさわしいグラフの番号を選べ。



1. 配列の中にデータが規則性が無い状態で保存してある。この配列の中に重複があるすべてのデータの件数を確認する処理。
2. C 言語の配列に先頭から隙間無く昇順にデータを保存してある。この配列に 1 件のデータを追加する処理。
3. ハッシュ表の大きさより N が小さいという条件で、ハッシュ表に 1 件新規のデータを追加する処理。
4. 2 分探索木の中から、指定された値のデータを検索し削除する処理。

2　データ構造設計 (× 20)

病院の入院患者のデータベースを作成したい。1 人あたりのデータは、部屋番号 (5 桁整数)、名前、生年月日 (8 桁整数)、診療科とする。病室は個室・複数人の場合がある。

このデータをどの様に管理すると良いか考え、そのデータ構造の宣言と、そのデータ構造のイメージが解るような図を示し、1 人の患者が入院してきた時の処理の概要を説明せよ。

データ例：

```
10001 斉藤徹　　19650207 内科
20015 山田太郎 19701203 外科
10103 鈴木花子 19780910 婦人科
20015 田中次郎 19610314 泌尿器科
  :      :          :      :
```

3　処理速度の予測 (× 20)

前設問の入院患者のデータベースにおいて、1 人が退院した時の処理を行う場合、あなたの設計したデータベースであれば、どの位の処理時間 T となるか？

全入院患者数を N とした場合、患者数が半分になった場合の処理時間 $T(N/2)$ を $T(N)$ を使って、理由と共に説明せよ。例えば「時間はおおよそ半分になる」といった表現でも良い。

4　2 分木の応用

0 と 1 の文字だけの文字列から、その 0/1 の並び順に応じて文字を出力するプログラムを以下の様に作った。2 つの設問に答えよ。

```
struct Tree {
  char* mes ;
  struct Tree* tree[2] ;
} ;
```

```
struct Tree* cons( char* s ,
                   struct Tree* t0 ,
                   struct Tree* t1 )
{ struct Tree* n =
     (struct Tree*)malloc( sizeof( struct Tree ) ) ;
  if ( n != NULL ) {
    n->mes = s ;
    n->tree[0] = t0 ;
    n->tree[1] = t1 ;
  }
  return n ;
}
```

★設問 (1) top の示すデータ構造を分かりやすく図示せよ
× 10

```
void convert( char s[] , struct Tree* top ) {
  struct Tree* p = top ;
  int i = 0 ;
  for( i = 0 ; s[ i ] != '\0' ; i++ ) {
    if ( p == NULL )
      p = top ;
    printf( "%s" , p->mes ) ;
    if ( s[i] == '0' )
      p = p->tree[0] ;
    else
      p = p->tree[1] ;
  }
```

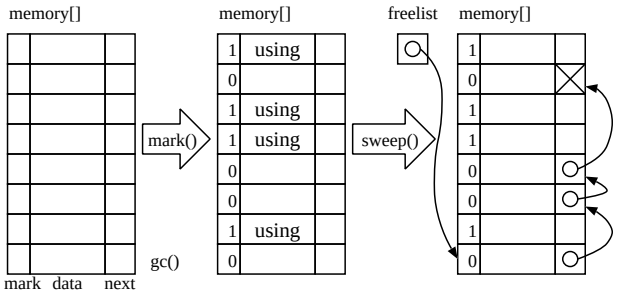
間違いの訂正：
"}"が抜けている。

```
void main() {
  struct Tree *top
    = cons( "a" ,
            cons( "b" , NULL , NULL ) ,
            cons( "c" ,
                  cons( "d" , NULL , NULL ) ,
                  cons( "e" , NULL , NULL ) ) ) ;
  convert( "0111" , top ) ;
}
```

★設問 (2) プログラムの表示内容を答えよ。
× 10

5　ガベージコレクタ

メモリ管理の理解のために、マーク＆スイープ法のガベージコレクタの処理を記述したい。貸し出すメモリは固定長で、下に示すような配列で、図で示すように、mark=1 が使用中とする。以下のプログラムの空白部を埋めよ。




```
#define CELL_SIZE 100

struct Cell {                          | void mark() {
  int  mark ;                          |   何らかの処理で使用中 Cell の mark を 1 にする。
  int  data ;                          | }
  struct Cell* next ;                  |
} ;                                    | void gc() { // ガベージコレクタ
                                       |   mark() ;  // 目印をつける
struct Cell memory[ CELL_SIZE ] ;      |   sweep() ; // 未使用を freelist に回収
struct Cell* freelist ;                | }

void sweep() {
  int i = 0 ;
  freelist =                  ;
             ~~~~~~~~~~~~~~~~(A)6
  for( i = 0 ; i < CELL_SIZE ; i++ ) {

    if (                            == 0 ) {
         ~~~~~~~~~~~~~~~~~~~~~~~~~~(B)7

      memory[ i ]->next =                  ;
                          ~~~~~~~~~~~~~~~~~(C)7
      freelist = memory + i ; // i 番目の Cell の場所
    }
  }
}
```

間違いの修正：
memory[i].next = ... ;

6　説明問題 (× 20)

参照カウンタ法について、具体的な例や図などを交えながら、その概要と利点・欠点などを説明せよ。