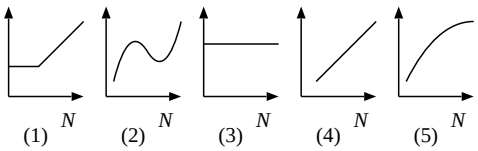
情報構造論 (斉藤)　後期期末試験　番号：
4EI　2008/02/07　(1/3)　氏名：

下記の 5 つの選択問題から 4 つを選んで解答せよ。

1　一般総括 (× 25)

1. 以下のようなデータ構造を採用した場合の処理時間やメモリ使用量は、データ件数 N に対してどの様に変化するか、最もふさわしい対応する変化のグラフを 1 つ選べ。 (3×3)
   (a) チェイン法によるハッシュの検索時間
   (b) 2 分木のメモリ使用量
   (c) あらかじめ昇順に並べられた配列から、効率よくデータを検索するための時間



2. 以下の説明にふさわしいデータ構造・方式の名称を答えよ。 (3×2)
   (a) 一般的なデータベースソフトで採用されているデータ構造
   (b) unix(OS) で、ハードリンクによる別名機能のあるファイルシステム管理で使われている方式

3. 2 分木で、 1000 件の処理時間が 100[msec] であった場合、 100,000 件の処理時間はどの程度となると予測されるか答えよ。 $(\times 10)$

2　データ構造の設計と分析 (× 25,8+9+8)

下記のような予定と日付のデータ構造を管理したい。
主な処理は、日付を指定して該当日の予定出力とする。

- 予定の対象者名 (最大 16byte の文字列)
- 予定年日付 (年, 月, 日の要素を持つこと)
- 予定の内容 (最大 64byte の文字列)

★例

```
| t-saitoh 2008 2  7 誕生日
| tomoko   2008 3 13 誕生日
| ayuka    2008 2 27 生活発表会
| t-saitoh 2008 2  1 青武台だより原稿
|  :
```

データ件数については、予定の対象者は最大 10 人、予定件数は予測できないものとする。あなたならこのデータ構造をどのように保存するか、設計せよ。
ただし、データ件数を N とした時の検索処理時間のオーダが $O(N)$ の手法は配点を 15(5+5+5) 点とする。

1. このデータを保存するために必要なデータ構造の宣言を示せ。
2. 保存・管理する方法を図などを交えて説明せよ。 (考え方を説明すれば良い)
3. あなたの処理方法で、データ件数を N とした時の、検索処理時間を根拠と共にオーダ記法で示せ。

3　参照カウンタ法 (× 25)

参照カウンタ法による下記のプログラムを作成した。
このデータのイメージ図と出力内容 (A) 〜 (E) を、
設問に応じて回答せよ。

```
struct StrList {
    int             refc ;
    char            data[ 20 ] ;
    struct StrList* next ;
} ;
struct StrList* scons( char s[] ,
                       struct StrList* n )
{   struct StrList* ans ;
    ans = (struct StrList*)malloc(
              sizeof( struct StrList ) ) ;
    if ( ans != NULL ) {
        ans->refc = 1 ;
        strcpy( ans->data , s ) ;
        ans->next = n ;
    }
    return ans ;
}
```

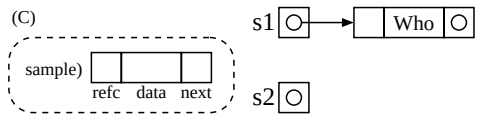
```
struct StrList* scopy( struct StrList* p )
{   if ( p != NULL )
        p->refc++ ;
    return p ;
}

void printStrList( struct StrList* p )
{
    for( ; p != NULL ; p = p->next )
         printf( "(%d)%s " , p->refc , p->data ) ;
}
```

```
void freeStrList( struct StrList* p )
{   if ( p != NULL ) {

        ~~~~~~~~~~~~~~~~~~~(A)
        if ( p->refc == 0 ) {

        }   ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~(B)
    }
}
void main() {
    struct StrList* s1 = scons( "Who" , scons( "are" , scons( "you" , NULL ) ) ) ;
    struct StrList* s2 = scons( "How" , scopy( s1->next ) ) ;
    /* (C) この時点でのデータ構造のイメージ図を答えよ */
    printStrList( s1 ) ; printf( "\n" ) ;  /* (D) 表示結果を答えよ */
    freeStrList( s1 ) ;
    printStrList( s2 ) ; printf( "\n" ) ;  /* (E) 表示結果を答えよ */
    freeStrList( s2 ) ;
}
```



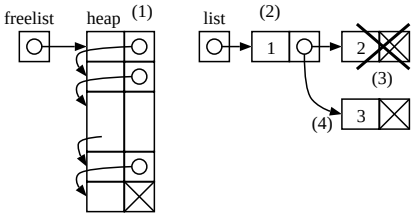
4　説明問題 (× 25)

1. 前設問のような方法が苦手とするデータ構造はどの様なものか、問題となるデータ構造の名称とその状況を図などを交えて具体的に説明せよ。(× 12)

2. 前設問のようなポインタ先で共有のあるデータの管理方法で、別の手法としてどういったものがあるか、名称や具体的な管理方法を説明せよ。(× 13)

5　ヒープ管理

ヒープメモリの管理を理解するために、以下のような自作のmy_malloc(),my_free() を作り、簡単なリスト処理を行う。
プログラムが正しく動くように、
下線部 (A) 〜 (E) にふさわしい処理を答えよ。




```
#define SIZE 10
struct List {
    int          data ;
    struct List* next ;
} ;
struct List  heap[ SIZE ] ;
struct List* freelist ;

void my_init() {
    int i ;
    freelist = &heap[0] ;
    for( i = 0 ; i < SIZE-1 ; i++ ) {

    }   ~~~~~~~~~~~~~~~~~~~~~~~~~~(A)

}   ~~~~~~~~~~~~~~~~~~~~~~~~~~(B)

void main() {
    struct List* list ;
    my_init() ;    /* (1) */

    list = cons( 1 , cons( 2 , NULL ) ) ; /* (2) */
    my_free( list->next ) ;               /* (3) */

    list->next = cons( 3 , NULL ) ;       /* (4) */
    my_free( list->next ) ;               /* (5) */
    my_free( list ) ;                     /* (6) */
}
```

```
struct List* my_malloc()
{   struct List* ans = freelist ;
    if ( freelist != NULL ) {

    }   ~~~~~~~~~~~~~~~~~~~~~~~~~~(C)
    return ans ;
}
void my_free( struct List* p ) {
    if ( p != NULL ) {

        ~~~~~~~~~~~~~~~~~~~~~~~~~~(D)

    }   ~~~~~~~~~~~~~~~~~~~~~~~~~~(E)
}
struct List* cons( int x , struct List* n )
{   struct List* ans ;
    if ( (ans = my_malloc()) != NULL ) {
        ans->data = x ;
        ans->next = n ;
    }
    return ans ;
}
```