

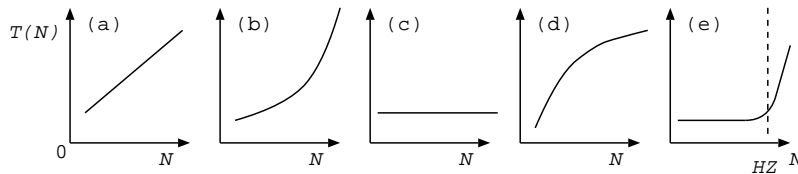
5つの選択問題より、4つを選んで回答せよ。

## 1 基本問題 (25)

1. あるアルゴリズムの処理時間のオーダーがデータ  $N$  件について  $O(\log N)$  で与えられたとする。この処理において、データ 100 件 で 1[msec] かかった場合、データが 10000 件 の場合の処理時間を答えよ。(9)

2. 以下の処理時間が、データ件数に対してどのように変化するか、図よりふさわしいものを選び。同じグラフを複数回答しても良い。(4x4)

- ・2分探索木に1件のデータを追加するのにかかる時間 ( )
- ・Deque(双方向キュー)の中から1件データを検索する時間 ( )
- ・選択法で配列を並べ替えるのにかかる時間 ( )
- ・チェイン法で新しく1件データを検索する時間 ( )



## 2 データ構造の設計 (25)

	開始アドレス	終了アドレス	ドメイン名
IPv4 アドレスが与えられたら、どの組織に割り当てられているのかを調べるためのデータベースを設計する。	192.156.145.0	192.156.149.255	fukui-nct.ac.jp
	64.33.0.0	64.33.63.255	ttn.ne.jp
	64.124.8.0	64.124.8.255	above.net

(例) 64.33.3.150 が与えられたら ttn.ne.jp を返したい。

IP アドレスは表の様に、割り当てられている領域の最初のアドレスと最後のアドレスとドメイン名から構成されているものとする。割り当てられているアドレスの最初と最後で、先頭 8bit が異なるような大きなアドレス空間をもつ組織はなく、データ件数は最大でも 1 万件とする。

データ件数  $N$  に対して、 $O(N)$  となるようなアルゴリズムは減点とする。

1. この処理のために必要なデータ構造の宣言を C 言語にて具体的に示せ。(10)
2. 前述のデータが保存されているイメージ図などを交えながら、アドレスから対応するドメイン名を見つけ出す処理について具体的に説明せよ。(15)

### 3 チェイン法のプログラム穴埋め (25)

誕生日と名前のデータをチェイン法で保存する。誕生日は、1965年2月7日といった日付は19650207 という整数値で扱う。

```

struct BNList {
    int    birthday ;
    char   name[ 16 ] ;
    struct BNList* next ;
} ;
| struct BNList* table[ 13 ] ; // 全部 NULL で初期化
| int hash_func( int b_day ) {
|     // b_day=19650207 => return 2
|     return ((b_day / 100) % 100) ;
| }

struct BNList* bn_cons( int bd , char* nm , struct BNList * nx ) {
    struct BNList* ans ;
    ans = (struct BNList*)malloc( sizeof( struct BNList ) ) ;
    if ( ans != NULL ) {
        ans->birthday = bd ;
        strcpy( ans->name , nm ) ;
        ans->next = nx ;
    }
    return ans ;
}

void entry( int bd , char* nm ) {
    int idx = hash_func( bd ) ;
    table[idx] = bn_cons( bd , nm , table[idx] ) ;
}

char* find( int bd ) {
    int idx = _____(A)
    for( struct BNList* p = _____ ;
        p != NULL ; _____ ) {
        if ( _____ )
            return _____ ;
    }
    return NULL ;
}

int main() {
    entry( 19650402 , "ueyama" ) ;
    entry( 19760303 , "tomoko" ) ;
    entry( 19650428 , "kondoh" ) ;
    /* (a) */
    printf( "%s\n" , find( 19760303 ) ) ;
    return 0 ;
}

```

[設問1] main() の (a) の時点での table[] のメモリ状態をイメージ図で表現せよ。(10)

[設問2] 誕生日をキーとして登録データを探す関数 find() の下線部 (A) ~ (E) を答えよ。(3x5=15)

### 4 説明問題 (25)

以下の3つの説明問題から2つを選んで回答せよ。(12+13)

1. 参照カウンタ法について説明せよ。このデータ構造が苦手とするデータを交えて説明すること。
2. 関数へのポインタについてプログラム例などを交えながら説明せよ。
3. ヒープメモリの断片化について説明せよ。

## 5 ガベージコレクタの挙動 (25)

```

struct List {
    int flag ;
    int data ;
    struct List* next ;
} ;

struct List heap[ 5 ] = {
    { 0 , -1 , heap + 1 } ,
    { 0 , -1 , heap + 2 } ,
    { 0 , -1 , heap + 3 } ,
    { 0 , -1 , heap + 4 } ,
    { 0 , -1 , NULL } ,
} ;

struct List* freelist = heap ;

struct List* my_malloc() {
    struct List* ans = freelist ;
    freelist = freelist->next ;
    return ans ;
}

void my_free( struct List* p ) {
    p->next = freelist ;
    freelist = p ;
}
    
```

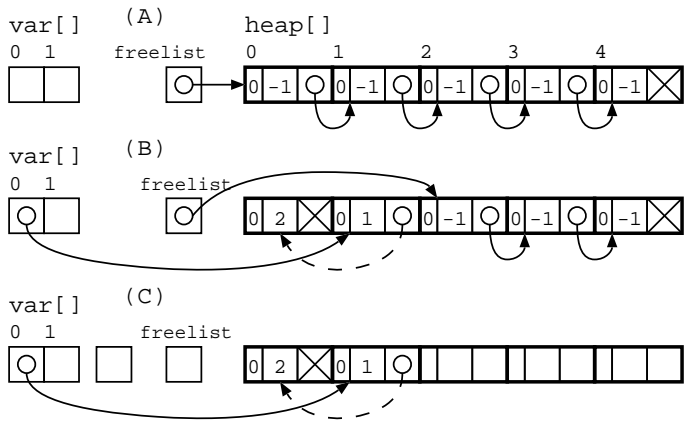
```

struct List* var[ 2 ] ;

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

```

void my_gc() {
    for( int i = 0 ; i < 2 ; i++ ) { // mark
        for( struct List* p = var[ i ] ;
            p != NULL ; p = p->next ) {
            if ( p->flag )
                break ;
            p->flag = 1 ;
            printf( "%d mark\n" , p->data ) ;
        }
    }
    /* (D) */
    for( int i = 0 ; i < 5 ; i++ ) { // sweep
        struct List* p = heap + i ;
        if ( !p->flag ) {
            my_free( p ) ;
            printf( "%d free\n" , p->data ) ;
        }
    }
    /* (E) */
}
    
```



```

int main() {
    struct List* p ;
    /* (A) */
    var[ 0 ] = my_cons( 1 , my_cons( 2 , NULL ) ) ;
    /* (B) */
    p = my_cons( 3 , my_cons( 4 , NULL ) ) ;
    var[ 1 ] = my_cons( 5 , NULL ) ;
    /* (C) */
    my_gc() ;
}
    
```

[ヒント]  
プログラムの初期状態 (A) と、  
var[0] への代入処理後 (B) の  
メモリのイメージ図を参考にせよ

[設問]  
プログラムを実行した時に  
表示される内容を答えよ。

ただし、(E) の状態のメモリのイメージ図を示すこと。