

令和 5 年度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No. 1

コース等	コンピュータ理工学 コース	試 験 分 野	専 門 科 目
------	------------------	---------	---------

試験時間は 1 時間 30 分です。試験監督から指示があるまで、この表紙をめくってはいけません。
次ページ以降に問題が、右上に番号付けされた用紙に分けて出題されています。配点は 140 点です。
解答にあたっては、解答用紙の表紙の指示に従いなさい。

解答開始の合図の後、各分野の問題について下表中に示す No. の用紙が綴じ込まれていることを確認しなさい。用紙に乱丁・落丁がある場合には、手を挙げて試験監督に知らせなさい。

	分野名	問題用紙の ページ番号
必須	アルゴリズムとデータ構造, 並びにプログラミング	No. 2~9

解答は、原則、1 問につき 1 枚を使用する。もしも解答用紙のスペースが不足の場合には、手を挙げて試験監督に知らせること。

すべての解答用紙について、受験番号欄に受験番号を記入の上で試験終了後に提出しなさい。本用紙を含むすべての問題用紙についても、回収します。

令和 5 年度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No 2

コース等	コンピュータ理工学 コース	試 験 分 野	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	---------	-----------------------------

問 1 以下の関数 MM1, MM2 は, 2 つの $2^n \times 2^n$ ($n \geq 0$) 行列 A, B の積 $C = A \times B$ を計算する.

```

1: function MM1( $A, B$ )
2:   if size of  $A$  and  $B = 1$  ( $=2^0$ ) then
3:     return  $A \times B$ 
4:   else
5:      $a_{11}, a_{12}, a_{21}, a_{22} = \text{divide}(A)$ 
6:      $b_{11}, b_{12}, b_{21}, b_{22} = \text{divide}(B)$ 
7:      $c_{11} = \text{MM1}(a_{11}, b_{11}) + \text{MM1}(a_{12}, b_{21})$ 
8:      $c_{12} = \text{MM1}(a_{11}, b_{12}) + \text{MM1}(a_{12}, b_{22})$ 
9:      $c_{21} = \text{MM1}(a_{21}, b_{11}) + \text{MM1}(a_{22}, b_{21})$ 
10:     $c_{22} = \text{MM1}(a_{21}, b_{12}) + \text{MM1}(a_{22}, b_{22})$ 
11:     $C = \text{combine}(c_{11}, c_{12}, c_{21}, c_{22})$ 
12:    return  $C$ 
13:   end if
14: end function

```

```

1: function MM2( $A, B$ )
2:   if size of  $A$  and  $B = 1$  ( $=2^0$ ) then
3:     return  $A \times B$ 
4:   else
5:      $a_{11}, a_{12}, a_{21}, a_{22} = \text{divide}(A)$ 
6:      $b_{11}, b_{12}, b_{21}, b_{22} = \text{divide}(B)$ 
7:      $M1 = \text{MM2}(a_{12} - a_{22}, b_{21} + b_{22})$ 
8:      $M2 = \text{MM2}(a_{11} + a_{22}, b_{11} + b_{22})$ 
9:      $M3 = \text{MM2}(a_{11} - a_{21}, b_{11} + b_{12})$ 
10:     $M4 = \text{MM2}(a_{11} + a_{12}, b_{22})$ 
11:     $M5 = \text{MM2}(a_{11}, b_{12} - b_{22})$ 
12:     $M6 = \text{MM2}(a_{22}, b_{21} - b_{11})$ 
13:     $M7 = \text{MM2}(a_{21} + a_{22}, b_{11})$ 
14:     $c_{11} = M1 + M2 - M4 + M6$ 
15:     $c_{12} = M4 + M5$ 
16:     $c_{21} = M6 + M7$ 
17:     $c_{22} = M2 - M3 + M5 - M7$ 
18:     $C = \text{combine}(c_{11}, c_{12}, c_{21}, c_{22})$ 
19:    return  $C$ 
20:   end if
21: end function

```

令和 5 年 度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No 3

コース等	コンピュータ理工学 コース	試 験 分 野	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	---------	-----------------------------

関数中の divide は $2^n \times 2^n$ 行列を以下のように 4 つの $2^{n-1} \times 2^{n-1}$ 行列に分割する.

$$M = \begin{pmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{pmatrix}$$

combine は divide の逆の操作を行う.

$2^n \times 2^n$ 行列に対する MM1 の乗算の回数を $m1(n)$, MM2 の乗算の回数を $m2(n)$ として, 以下の設問に答えなさい.

- (1) $m1(0)$ および $m1(1)$ はいくつか答えなさい.
- (2) 「 $m1(n) = \boxed{A} m1(n-1)$.」左の式の空欄 A に当てはまる整数を答えなさい.
- (3) 「(1), (2)より, すべての $n \geq 0$ に対して $m1(n) = \boxed{B}$ が得られる。」左の式の空欄 B に当てはまる n の関数を答えなさい.
- (4) $N = 2^n$ とするとき, (3)で導出した MM1 の乗算の回数を N の関数として表しなさい.
- (5) 「すべての $n \geq 0$ に対して $m2(n) = \boxed{C}$ である。」左の式の空欄 C に当てはまる n の関数を答えなさい.
- (6) $N = 2^n$ とするとき, (5)で導出した MM2 の乗算の回数を N の関数として表しなさい.
- (7) 2つのアルゴリズムの実際の実行時間を, 行列の加減算の回数と再帰呼び出しの回数に着目して, 行列サイズが小さい場合と大きい場合について予想しなさい.

令和 5 年度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No. 4

コース等	コンピュータ理工学 コース	試験分野	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	------	-----------------------------

問 2 図 1 は要素数が 4 であるヒープの例を表しており, ソースファイル `heap.cpp` では配列によってヒープを表現している. 以下の設問に答えなさい.

- (1) 2 分木がヒープであるための条件を書きなさい.
- (2) 図 1 に示すヒープに, 新たに要素 ⑤ を追加するときに, ヒープがどのように変化していくかを図示しなさい.
- (3) (2) で完成したヒープに, 新たに要素 ③ を追加するときに, ヒープがどのように変化していくかを図示しなさい.
- (4) (3) で完成したヒープから, 最小の要素を 1 つ取り出すときに, ヒープがどのように変化していくかを図示しなさい.

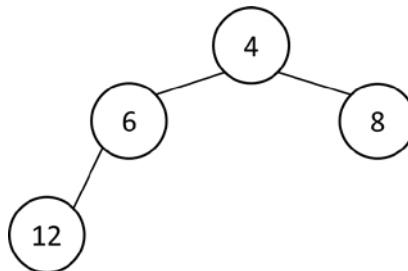


図 1 ヒープの例

- (5) ソースファイル `heap.cpp` の空欄 から に当てはまるコードを記述しなさい.
- (6) `heap.cpp` をコンパイル・実行したときの出力結果を示しなさい.
- (7) ヒープソートでは, n 個の要素の列をヒープに入れて, 要素を 1 つずつ取り出すことによって, ソーティングを実現する. 以下の計算量のオーダーを答えなさい.
 - (a) n 個の要素を持つヒープに, 新たに要素を 1 つ追加する.
 - (b) n 個の要素を持つヒープから, 要素を 1 つ取り出す.
 - (c) n 個の要素の列に対してヒープソートを行う.

令和 5 年 度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No 5

コース等	コンピュータ理工学 コース	試 験 分 野	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	---------	-----------------------------

heap.cpp

```
#include <iostream>
using namespace std;
#define MAX_ARRAY_SIZE 128

class heap {
    int a[MAX_ARRAY_SIZE];
    int n;

public:
    heap(){ n = 0; };
    void Enqueue(int data);
    int DequeueMin();
};

// ヒープに要素を1つ挿入する
void heap::Enqueue(int data){
    int parent; // 親頂点
    int i = ; // 要素を挿入する頂点

    while (i>0){
        parent = ;
        if( a[parent] > data ){
            a[i] = a[parent];
            i = parent;
        }else{
            break;
        }
    }
    a[i] = data;
    n++;
}

// ヒープから最小の要素を取り出す
int heap::DequeueMin(){
    if( n <= 0 ){ return -1; }

    int mindata = a[0];
    int target = ; // 優先的に移動の対象となる要素
    n--;

    // 次ページへつづく
```

令和 5 年度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No. 6

コース等	コンピュータ理工学 コース	試験分野	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	------	-----------------------------

```
int i = ; // 探索を開始する頂点
int left = ;
int right = ;

while( left < n ){
    int smaller = left;
    if( right < n & a[right] < a[left] ){
        smaller = right;
    }
    if( target > a[smaller] ){
        a[i] = a[smaller];
        i = smaller;
    }else{
        break;
    }
    left = ;
    right = ;
}
a[i] = target;
return mindata;
}

int main(){
    heap h;
    int a[4] = {12, 6, 8, 4};

    for(int i=0; i<4; i++){
        h.Enqueue(a[i]);
    }
    h.Enqueue(5);
    h.Enqueue(3);

    int k = h.DequeueMin();
    while (k >= 0){
        cout << k << endl;
        k = h.DequeueMin();
    }
    cout << endl;

    return 0;
}
```

令和 5 年度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No. 7

コース等	コンピュータ理工学 コース	試験分野	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	------	-----------------------------

問 3 No 8 ページから No 9 ページの囲みの中の C++言語のソースコードは、1 画素の位置 (Point クラスのデータメンバ x, y) と三原色 (R, G, B クラスのデータメンバ r, g, b (いずれも 0 から 255 までの整数をとる.)) を扱うコードである. このコードについて、以下の設問に答えなさい.

- (1) コンパイル・実行したとき、`main()`関数内の//A の行のコードの実行によって、以下の<実行結果>の囲みのとおり、各画素の位置とカラーコード (#のあとに `rgb` の順で 16 進数表記) が標準出力へ出力される. この<実行結果>となるよう、Pixel クラスのメンバ関数 `unsigned long code()` の定義内の空欄 `X` に当てはまる適切なコードを記述しなさい.

<実行結果>

```
(011,022)#ff0010
(033,044)#ffff00
(055,066)#002080
```

- (2) 設問(1)の題意を満たすコードが記述されたことを前提として、コンパイル・実行したときの、`main()`関数内の//B の行のコードの実行による標準出力への出力をすべて示しなさい.
- (3) クラス R, G, B の定義部分の//C の行の `virtual` の部分を削除してコンパイルすると、コンパイルエラーとなる. この理由を 100 字程度 (または 30-50 words) で説明しなさい.
- (4) `main()`関数内の//D の行の `delete[]`の代わりに `delete` と書くと、コンパイルして実行できる場合もあるが、実行時に問題が生じる可能性がある. どのような問題が生じる可能性があるのかについて、80 字程度 (または 20-40 words) で説明しなさい.

令和 5 年 度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No 8

コース等	コンピュータ理工学 コース	試 験 分 野	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	---------	-----------------------------

```
#include <iostream>
#include <iomanip>
using namespace std;

class Point {
public:
    int x;
    int y;
    Point(int xx = 0, int yy = 0) :x(xx), y(yy) {}
};

class R : virtual public Point { //C
public:
    int r;
    R(int rr = 0) :r(rr) {}
};

class G : virtual public Point { //C
public:
    int g;
    G(int gg = 0) :g(gg) {}
};

class B : virtual public Point { //C
public:
    int b;
    B(int bb = 0) :b(bb) {}
};

class Pixel : public R, public G, public B {
public:
    Pixel(Point pp = Point(10, 10), int rr = 10, int gg = 10, int bb = 10)
        :Point(pp), R(rr), G(gg), B(bb) {}
    ~Pixel() { cout << "destruction\n"; }
    unsigned long code() { return ; }
};
```

//次ページへつづく

令和 5 年 度
山梨大学 大学院医工農学総合教育部 修士課程 工学専攻

入 学 試 験 問 題

No. 9

コース等	コンピュータ理工学 コース	試験分野	アルゴリズムとデータ構造, 並びにプログラミング
------	------------------	------	-----------------------------

```
//前ページからのつづき

void showPixel(Pixel p[], int n)
{
    for (int i = 0; i < n; ++i) {
        cout << setfill('0');
        cout << dec << '(' << setw(3) << p[i].x << ','
            << setw(3) << p[i].y << ')';
        cout << '#' << setw(6) << hex << p[i].code() << '\n';
    }
}

int main()
{
    int N = 3;
    Pixel* p;
    p = new Pixel[N];
    showPixel(p, N); //B

    p[0] = Pixel(Point(11, 22), 255, 0, 16);
    p[1] = Pixel(Point(33, 44), 255, 255, 0);
    p[2] = Pixel(Point(55, 66), 0, 32, 128);
    showPixel(p, N); //A

    delete[] p; //D

    return 0;
}
```