

Hw 1.

$\Theta \Rightarrow$ upper and lower bounds
 $O \Rightarrow$ upper bound only

2.(a) $\sum_{i=0}^n i^2 \stackrel{?}{=} \Theta(n^3)$

$$\sum_{i=0}^n i^2 = \sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6} = \frac{1}{6}(2n^3 + 3n^2 + n)$$

$$\frac{1}{6}n^3 \leq \frac{1}{6}(2n^3 + 3n^2 + n) \leq 2n^3 \text{ when } n \rightarrow \infty, \text{ thus } \sum_{i=0}^n i^2 = \Theta(n^3)$$

(b) $n! \stackrel{?}{=} O(n^n)$

$$n! = n(n-1)(n-2)(n-3) \cdots 3 \cdot 2 \cdot 1 < n \cdot n \cdot n \cdot n \cdots n \cdot n \cdot n = n^n$$

Thus according to the definition of $O(f(x))$, $n! = O(n^n)$

3.(a) $10n^2 + 9 \stackrel{?}{=} O(n)$

In the long run, $n \leq 10n^2 + 9 \leq 11n^2$ $\xrightarrow{\text{upper bound}}$

Thus according to the definition of $O(f(x))$, $10n^2 + 9 = O(n^2)$
not $O(n)$

(b) $\frac{n^2}{\log n} \stackrel{?}{=} \Theta(n^2)$

$$\frac{n^2}{2 \log n} < \frac{n^2}{\log n} < \frac{3n^2}{\log n} \text{ and there does not exist a constant } k$$

such that $kn^2 \leq \frac{n^2}{\log n} \leq 3n^2$ as $n \rightarrow \infty$. Therefore the big theta for $\frac{n^2}{\log n}$ is $\Theta(\frac{n^2}{\log n})$ not $\Theta(n^2)$

7. A * B * C in prefix form

$$A * B * C = \underbrace{(A * B)}_{\text{A * B}} * C = ** ABC$$

```
#include <iostream>
using namespace std;
int main(void){
    int arr[10]={0, 1, 2, 3, 4, 0, 6, 7, 8, 0};
    int n = sizeof(arr)/sizeof(arr[0]);
    cout << "The first number of the array appears " << count(arr, arr+n, arr[0]) << " time(s).\n";
}

```

The first number of the array appears 3 time(s).
Program ended with exit code: 0

| (a).

A Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help Computer Science Computer Science My Mac Mon Mar 21 2:27 AM

Computer Science C matrix transpose Computer Science C matrix transpose main() matrix transpose main()

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

void transpose(int mx[3][4], int a, int b){
    int nmx[1000][1000]=();
    for(int i=0; i<a; i++){
        for(int j=0; j<b; j++){
            nmx[j][i]=mx[i][j];
        }
    }
    cout << "\nThe transposed matrix: \n";
    for(int i=0; i<b; i++){
        for(int j=0; j<a; j++){
            cout << nmx[i][j] << " ";
        }
        cout << "\n";
    }
}

int main(void){
    int mx[3][4]={ {1, 2, 3, 4},
                   {3, 4, 5, 6},
                   {5, 6, 7, 8} };
    transpose(mx, 3, 4);
    return 0;
}
```

The original matrix:
1 2 3 4
3 4 5 6
5 6 7 8

The transposed matrix:
1 3 5
2 4 6
3 5 7
4 6 8

Program ended with exit code: 0

Line: 20 Col: 16

All Output: Filter

| (b) .

A screenshot of an Xcode project titled "Computer Science" with a target named "complex number". The code editor shows a C++ file named "complex number" containing the following code:

```
1 #include <iostream>
2
3 using namespace std;
4
5 class Complex { // The class
6 public: // Access specifier
7     Complex(): r(0), i(0){};
8     int r;
9     int i;
10 };
11
12 int main() {
13     Complex n1; // Create an object of MyClass (this will call the constructor)
14     Complex n2;
15     n2.r=3;
16     n2.i=1;
17     cout << "The default number: " << n1.r << "+" << n1.i << "i\n";
18     cout << "The modified number: " << n2.r << "+" << n2.i << "i\n";
19
20     return 0;
21 }
```

The output window at the bottom displays the following text:

```
The default number: 0+0i
The modified number: 3+1i
Program ended with exit code: 0
```

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help Computer Science Computer Science My Mac Mon Mar 21 2:30 AM

```
Computer Science quad main()
1 #include <iostream>
2 using namespace std;
3
4 class Quad {
5 public:
6     Quad(): x2(0), x1(0), x0(0){}
7     int x2;
8     int x1;
9     int x0;
10
11    int add(int a, int b){
12        return a+b;
13    }
14 }
15
16 int main(void){
17     Quad q1;
18     Quad q2;
19
20     q1.x2=3;
21     q1.x1=2;
22     q1.x0=1;
23     q2.x2=6;
24     q2.x1=4;
25     q2.x0=2;
26
27     cout << "The first function: " << q1.x2 << "x^2 + " << q1.x1 << "x + " << q1.x0 << "\n";
28     cout << "The second function: " << q2.x2 << "x^2 + " << q2.x1 << "x + " << q2.x0 << "\n";
29     cout << "The added function: " << q1.add(q1.x2, q2.x2) << "x^2 + " << q1.add(q1.x1, q2.x1) << "x + " << q1.add(q1.x0, q2.x0) << "\n";
30
31    return 0;
32 }
```

The first function: $3x^2 + 2x + 1$
The second function: $6x^2 + 4x + 2$
The added function: $9x^2 + 6x + 3$
Program ended with exit code: 0

All Output 5 Line: 29 Col: 104



```

 1 #include <stdio.h>
2 #include <iostream>
3 using namespace std;
4
5 class Bag{
6 protected:
7     int *array;
8     int capacity;
9     int top;
10
11 public:
12     Bag (int bagCapacity = 10);
13     ~Bag(){}
14     int Size() const{
15         return capacity;
16     }
17     bool IsEmpty() const{
18         return !capacity;
19     }
20     int Element() const{
21         return 0;
22     }
23     void Push(const int){}
24     void Pop(){}
25 };
26
27 template <class Tpl>
28 class Queue : public Bag{
29 public:
30     Queue(int queueCapacity = 20):
31         Bag(queueCapacity){
32         queue = new Tpl[queueCapacity]{0};
33         length = 1;
34     };
35
36     bool IsEmpty() const {
37         return !length;
38     };
39     int Size() const{
40         return length;
41     }
42     Tpl& Front() const{
43         return *queue;
44     }
45     Tpl& Rear() const{
46         return *(queue+length-1);
47     }
48     void Push(const Tpl& item){
49         length++;
50         *(queue+length-1) = item;
51     }
52     void Pop(){
53         for(int i=0; i<length-1; i++){
54             *(queue+i) = *(queue+i+1);
55         }
56         length--;
57     }
58     void Print(){
59         for(int i=0; i<length; i++){
60             cout << *(queue+i) << "\t";
61         }
62         cout << endl;
63     }
64     private:
65     Tpl *queue;
66 };
67
68
69
70 int main(){
71     Queue<int> que;
72     que.Print();
73     cout << "Rear: " << que.Rear() << endl;
74     cout << "Front: " << que.Front() << endl;
75     cout << "IsEmpty: " << que.IsEmpty() <<
76         endl;
77     cout << "Size: " << que.Size() << endl <<
78         endl;
79     que.Push(1);
80     que.Push(2);
81     que.Pop();
82     que.Push(3);
83     que.Print();
84     cout << "Rear: " << que.Rear() << endl;
85     cout << "Front: " << que.Front() << endl;
86     cout << "IsEmpty: " << que.IsEmpty() <<
87         endl;
88     cout << "Size: " << que.Size() << endl <<
89         endl;
90     que.Pop();
91     que.Pop();
92     que.Pop();
93     que.Print();
94     cout << "IsEmpty: " << que.IsEmpty() <<
95         endl;
96     cout << "Size: " << que.Size() << endl <<
97         endl;
98     return 0;
99 }
100

```

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help Computer Science My Mac Mon Mar 21 8:56 PM

Computer Science > 大-TC++ > queue Queue

```
87     que.Push(3);
88     que.Print();
89     cout << "Rear: " << que.Rear() << endl;
90     cout << "Front: " << que.Front() << endl;
91     cout << "IsEmpty: "<< que.IsEmpty() << endl;
92     cout << "Size: " << que.Size() << endl << endl;
93
94     que.Pop();
95     que.Pop();
96     que.Pop();
97     que.Print();
98     cout << "IsEmpty: "<< que.IsEmpty() << endl;
99     cout << "Size: " << que.Size() << endl << endl;
100
101    return 0;
102 }
```

Line: 34 Col: 7

```
0
Rear: 0
Front: 0
IsEmpty: 0
Size: 1

1 2 3
Rear: 3
Front: 1
IsEmpty: 0
Size: 3

IsEmpty: 1
Size: 0

Program ended with exit code: 0
```

All Output: 6.



```

1 #include <iostream>
2 using namespace std;
3 int a=0;
4
5 class Matrix{
6 private:
7     int** matrix; // 2D array
8     int X; // Matrix rows
9     int Y; // Matrix columns
10 public:
11     // Default Constructor
12     Matrix(){
13         X = 0;
14         Y = 0;
15         matrix = NULL;
16     }
17     // Constructor with parameters
18     Matrix(int _X, int _Y, int mx[4][7]){
19         int sparse[3][10]={0};
20
21         X = _X;
22         Y = _Y;
23
24         for(int i=0; i<Y; i++){
25             for(int j=0; j<X; j++){
26                 if(mx[i][j]==0) {
27                     sparse[0][a]=mx[i][j];
28                     sparse[1][a]=i; // Row
29                     sparse[2][a]=j; // Column
30                     a++;
31                 }
32             }
33         }
34
35         // Create memory space for the matrix
36         matrix = new int*[X];

```



```

37             for (int i=0; i<X; i++) matrix[i] =
38                 new int[Y];
39
40             // Put the sparse matrix into the
41             // newly created matrix
42             for (int i=0; i<X; i++){
43                 for (int j=0; j<Y; j++){
44                     matrix[i][j] = sparse[i][j];
45                 }
46             }
47
48             // Copy constructor
49             Matrix(const Matrix& N)
50             {
51                 X = N.X;
52                 Y = N.Y;
53
54                 matrix = new int*[X];
55                 for (int i=0; i<X; i++) matrix[i] =
56                     new int[Y];
57
58                 for (int i=0; i<X; i++){
59                     for (int j=0; j<Y; j++){
60                         matrix[i][j] = N.matrix[i][j];
61                     }
62                 }
63
64                 void showSpa(){
65                     for (int i=0; i<X; i++){
66                         for (int j=0; j<Y; j++){
67                             cout << matrix[i][j] << "\t";
68                         }
69                         cout << endl << endl;
70                     }
71
72                 void showReg(){
73                     int mx[10][10]={0};
74                     for(int i=0; i<X; i++){
75                         mx[matrix[i][0]][matrix[i][1]] =
76                             [i]=matrix[i][0];
77                     }
78                     for(int i=0; i<X; i++){
79                         for(int j=0; j<Y; j++){
80                             cout << mx[i][j] << "\t";
81                         }
82                         cout << endl << endl;
83                     }
84
85                 int main(void){
86                     int mx[4][7]={0, 2, 0, 0, 0, 0, 1,
87                                 {0, 0, 0, 0, 3, 0, 0},
88                                 {0, 0, 0, 5, 0, 0, 0},
89                                 {7, 0, 0, 0, 0, 0, 0}};
90
91                     int a=sizeof(mx)/sizeof(mx[0]);
92                     int b=sizeof(mx[0])/sizeof(mx[0][0]);
93
94                     cout << "The original matrix in regular
95                     matrix representation:" << endl;
96                     for(int i=0; i<X; i++){
97                         for(int j=0; j<Y; j++){
98                             cout << mx[i][j] << "\t";
99                         }
100                        cout << endl << endl;
101                    }
102
103

```

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help

Computer Science Computer Science My Mac Mon Mar 21 8:32 PM

copy constructor

```
Computer Science [C:\] copy constructor Matrix(_X,_Y)
100 }
101 cout << endl << endl;
102 }
103
104 Matrix x(a, b, mx);
105 cout << "The original matrix in sparse matrix representation:" << endl;
106 x.showSpa();
107
108 Matrix y(x);
109 cout << "The copied matrix in sparse matrix representation:" << endl;
110 y.showSpa();
111
112 cout << "The copied matrix in regular matrix representation:" << endl;
113 y.showReg();
114
115 return 0;
116 }
117
```

Line: 36 Col: 5



```
The original matrix in regular matrix representation:  
0 2 0 0 0 0 1  
0 0 0 0 3 0 0  
0 0 0 5 0 0 0  
7 0 0 0 0 0 0  
  
The original matrix in sparse matrix representation:  
2 1 3 5 7  
0 0 1 2 3  
1 6 4 3 0  
  
The copied matrix in sparse matrix representation:  
2 1 3 5 7  
0 0 1 2 3  
1 6 4 3 0  
  
The copied matrix in regular matrix representation:  
0 2 0 0 0 0 1  
0 0 0 0 3 0 0  
0 0 0 5 0 0 0  
7 0 0 0 0 0 0  
  
Program ended with exit code: 0
```

The computing time of the copy constructor depends on the number of non-zero elements in the sparse matrix.

$\rightarrow O(n)$, $n = \#$ of non-zero elements



Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help Computer Science > 大-TC++ > insertion sort main() Computer Science > My Mac Finished running Computer Science : Computer Science ▲

```
1 #include <iostream>
2 using namespace std;
3
4 int main(void){
5     int arr[5] = {3, 2, 4, 1, 5};
6     cout << "Unsorted list:\n";
7     for(int i=0; i<5; i++) cout << arr[i] << " ";
8
9     for(int i=1; i<5; i++){ //starting from the 2nd item in the array
10        int temp=arr[i]; //save the item to be compared
11        int j = i-1; //start comparing from the item before i
12        while(j>=0 && temp>arr[j]){
13            arr[j+1]=arr[j]; //scooch the bigger num over one place
14            j-=1; //search the next item in line
15        }
16        arr[j+1]=temp; //put in the saved item after the smaller num
17    }
18
19    cout << "\nSorted list:\n";
20    for(int i=0; i<5; i++) cout << arr[i] << " ";
21    cout << "\n";
22
23    return 0;
24 }
```

Unsorted list:
3 2 4 1 5
Sorted list:
1 2 3 4 5
Program ended with exit code: 0

All Output:

A Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help Mon Mar 21 2:42 AM

Computer Science > 大一C++> Palindrome 2) No Selection

```
1 #include <iostream.h>
2 #include <iostream>
3 using namespace std;
4
5 int main(void){
6     char s[100]={0};
7     while(cin >> s){
8         int flag=1;
9         int len=strlen(s);
10        for(int i=0; i<len; i++){
11            if(s[i]!=s[len-i-1]) flag=0;
12        }
13        if(flag) cout << s << " -- is a palindrome.\n";
14        else cout << s << " -- is not a palindrome.\n";
15    }
16    return 0;
17 }
18
19 
```

Implicit conversion loses integer precision: 'size_t' (aka 'unsigned long') to 'Int'

level -- is a palindrome.
abs -- is not a palindrome.
civic
civic -- is a palindrome.
liiy
liiy -- is not a palindrome.

All Output : / 0.

