2021 2 학기

자료구조개론

기말고사

문제지

1. You will get +1 point for each correct answer.

2. Assume that each program includes proper header files such as stdio.h and math.h.

3. All the variables and arrays are properly initialized by 0 at the beginning.

Q1. Read the following program, and answer the questions.

```
void Sort(int *arr, int len)
{
    int tmp = 0;
    for(int i = len - 1; i >= 0; i--){
        for(int j = 0; j < i; j++){
            if(Ⓐ){
                tmp = arr[j];
                Ⓑ
                arr[j + 1] = tmp;
            }
        }
    }
}
```

```
int main(void){
    int n = 10;

    int a[10] = {9, 11, 4, 2, 3, 1, 5 ,7 ,10, 6};

    // Sort ascending order
    Sort(a, n);

    for(int i = 0; i < n; i++){
        printf("%d₩n", a[i]);
    }
}
```

1. What is time complexity of the sort program? (n is the number of items in the list)

   ① $\Theta(1)$ ② $\Theta(n)$ ③ $\Theta(\log_2 n)$
   ④ $\Theta(n\log_2 n)$ ⑤ $\Theta(n^2)$

2. What is the name of this sort algorithm?

   ① Selection sort
   ② Insertion sort
   ③ Bubble sort
   ④ Quick sort
   ⑤ Merge sort

3. What is the value of a[4] after sort?

   ① 1 ② 3 ③ 5 ④ 7 ⑤ 9

4. What is the correct statement at Ⓐ?

   ① a[j+1] < a[j];
   ② a[j+1] > a[j];
   ③ a[j+1] == a[j];
   ④ a[j-1] > a[j];
   ⑤ a[j-1] < a[j];

5. What is the correct statement at Ⓑ?

   ① a[j] = a[j+1];
   ② a[j] = a[j-1];
   ③ a[j+1] = a[j];
   ④ a[j-1] = a[j];
   ⑤ a[j+1] = a[j-1];

Q2. Given static hash with open addressing, bucket size 17 and slot size 1, answer the questions

Index	Key
0	
1	Ⓐ
2	
3	
4	
5	Ⓑ
6	
7	
8	Ⓒ
9	
10	
11	
12	Ⓓ
13	
14	
15	
16	

Hash descriptions

- Static hashing
- Bucket size 17
- Slot size 1
- Open addressing
- Linear probe
- Empty slots have NULL

Hash functions

H1(key)

⇨ **key % 17**

H2(key)

⇨ **(key * key) % 17**

Insert following keys in order

17
24
29
0
3
4
87
54
15
35
12
48
65

6. After insertion, what is the value at Ⓐ when hash function is H1

① NULL ② 17 ③ 0 ④ 65 ⑤ 48

7. After insertion, what is the value at Ⓓ when hash function is H1

① NULL ② 4 ③ 48 ④ 35 ⑤ 29

8. After insertion, what is the value at Ⓑ when hash function is H1

① NULL ② 0 ③ 15 ④ 54 ⑤ 87

9. After insertion, what is the value at Ⓑ when hash function is H2

① NULL ② 0 ③ 17 ④ 15 ⑤ 48

10. After insertion, what is the value at Ⓓ when hash function is H2

① NULL ② 48 ③ 65 ④ 35 ⑤ 87

11. After insertion, what is the value at Ⓒ when hash function is H2

① NULL ② 24 ③ 29 ④ 17 ⑤ 54

Q3. Read the following program, and answer the questions.

```
#define MAX_SIZE 10

void fct1(int list1[], int list2[], int i, int m, int n)
{
    int j,k,t;
    j = m + 1;
    k = i;
    while(i <= m && j <= n) {
        if (list1[i] <= list1[j]){
            list2[k++] = list1[i++];
        }
        else{
            list2[k++] = list1[j++];
        }
    }

    Ⓐ

    if (i > m){
        for(t = j; t <= n; t++){
            list2[t] = list1[t];
        }
    }
    else{
        for(t = i; t <= m; t++){
            list2[k+t-i] = list1[t];
        }
    }
}

void fct2(int list1[], int list2[], int n, int s)
{
    int i ;
    int j ;

    for (i = 0; i <= n - 2 * s + 1; i += 2 * s){
        fct1(list1, list2, i, i + s - 1, i + 2 * s - 1);
    }

    Ⓑ

    if((i + s - 1) < n){
        fct1(list1, list2, i, i + s - 1, n);
    }
    else{
        for(j=i; j <= n; j++){
            list2[j] = list1[j];
        }
    }
}
```

```
void Sort(int a[], int n)
{
    int s = 1;
    int extra[MAX_SIZE];

    while (s<n) {
        fct2(a, extra, n, s);
        s *= 2;

        Ⓒ

        fct2(extra, a, n, s);
        s *= 2;

        Ⓓ

    }
}

int main(void)
{
    int arr[MAX_SIZE] = {13, 26, 41, 72, 23, 1, 0, 65, 32, 55};

    Sort(arr, MAX_SIZE - 1);
}
```

12. What is the name of this sorting algorithm?

① Merge Sort ④ LSD Radix Sort

② Heap Sort ⑤ List Sort

③ MSD Radix Sort

13. What is the time complexity of this algorithm?

① $\Theta(1)$ ② $\Theta(n)$ ③ $\Theta(\log_2 n)$

④ $\Theta(n\log_2 n)$ ⑤ $\Theta(n^2)$

14. How many times fct1 is called?

① 8 ② 9 ③ 10 ④ 11 ⑤ 12

15. How many times fct2 is called?

① 3 ② 4 ③ 5 ④ 6 ⑤ 7

16. When fct1 is called twice, what is the value of list1[4] at Ⓐ?

① 26 ② 23 ③ 1 ④ 0 ⑤ 55

17. When fct1 is called 5 times, what is the value of list2 [8] at Ⓐ?

① 32 ② 72 ③ 26 ④ 13 ⑤ 65

18. When fct2 is called for the first time, what is the value of list1[3] at Ⓑ?

① 26 ② 32 ③ 72 ④ 41 ⑤ 55

19. When fct2 is called 3 times, what is the value of list1[5] at Ⓑ?

① 41 ② 0 ③ 1 ④ 13 ⑤ 23

20. When s is 2, what is the value of a[3] at Ⓒ?

① 26 ② 32 ③ 41 ④ 72 ⑤ 13

21. When s is 4, what is the value of a[5] at Ⓓ?

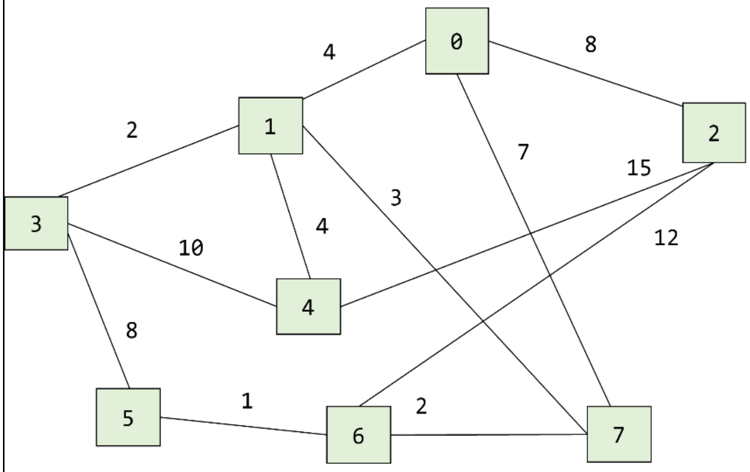
① 0 ② 23 ③ 1 ④ 55 ⑤ 65

Q4. Read the following graph program, and answer the questions. **Assume "func" is called with v = 0**.

```
#define VERY_LARGE_NUMBER 1000000
#define MAX_VERTICES 8

typedef struct __edge{
        int x;
        int y;
}edge;

typedef struct __treeData{
        edge T[MAX_VERTICES * MAX_VERTICES];
        int top;
}treeData;

void init_treeData(treeData *tree)
{
        edge tmp = {-1, -1};
        for(int i = 0; i < MAX_VERTICES * MAX_VERTICES; i++)
                tree->T[i] = tmp;
        tree->top = 0;
}
void fct(int graph[MAX_VERTICES][MAX_VERTICES]);

void push_to_T(treeData *tree, int x, int y)
{
        edge tmp;
        for(int i = 0; i < tree->top; i++){
                tmp = tree->T[i];
                if((tmp.x == x && tmp.y == y) || (tmp.x == y && tmp.y == x)){
                        return;
                }
        }
        tree->T[tree->top].x = x;
        tree->T[tree->top].y = y;

        tree->top++;
}
```

```
void fct(int graph[MAX_VERTICES][MAX_VERTICES])
{
        int no_edge = 0;

        treeData tree;
        int TV[MAX_VERTICES] = {0};

        init_treeData(&tree);

        TV[0] = true;

        while(no_edge < MAX_VERTICES - 1) {
                int min = VERY_LARGE_NUMBER;
                edge tmp = {0, 0};

                for (int i = 0; i < MAX_VERTICES; i++) {
                        if (TV[i]) {
                                for (int j = 0; j < MAX_VERTICES; j++) {
                                        if (!TV[j] && graph[i][j] != -1) {
                                                if (min > graph[i][j]) {
                                                        min = graph[i][j];

                                                        push_to_T(&tree, i, j);

                                                        tmp.x = i;
                                                        tmp.y = j;

                                                          Ⓐ

                                                }
                                        }
                                }
                        }
                }
                TV[tmp.y] = true;
                no_edge++;
        }
}
```

```
int main(void)
{
/* -1 implies there is no connection between the vertices. */
        int graph[MAX_VERTICES][MAX_VERTICES] = {
                { 0,   4,   8, -1, -1, -1, -1,   7},
                { 4,   0, -1,   2,   4, -1, -1,   3},
                { 8, -1,   0, -1, 15, -1, 12, -1},
                {-1,   2, -1,   0, 10,   8, -1, -1},
                {-1,   4, 15, 10,   0, -1, -1, -1},
                {-1, -1, -1,   8, -1,   0,   1, -1},
                {-1, -1, 12, -1, -1,   1,   0,   2},
                { 7,   3, -1, -1, -1, -1,   2,   0}};

        fct(graph);

        return 0;
}
```



22. What is the name of this graph algorithm?

① Kruskal's Algorithm

② Prim's Algorithm

③ Dijkstra's Algorithm

④ Bellman-Ford Algorithm

⑤ Floyd-Warshall Algorithm

23. Find the edge that is in the MST made by the above code.

① (4, 2) ② (1, 4) ③ (5, 3) ④ (3, 4) ⑤ (0, 7)

24. What is tree.top after the fct ended?

① 6 ② 7 ③ 8 ④ 9 ⑤ 10

25. Find the edge that cannot be in the tree.T when tree.top is 3 at Ⓐ.

① (0, 1) ② (0, 7) ③ (1, 4) ④ (1, 7) ⑤ (6, 5)

26. Find the edge that cannot be in the tree.T when tree.top is 6 at Ⓐ.

① (0, 1) ② (0, 2) ③ (1, 7) ④ (6, 5) ⑤ (3, 4)

27. Suppose that node 1 and node 7 are not connected.
Find the edge that is in the MST made by the above code and given assumption.

① (0, 1) ② (2, 6) ③ (5, 3) ④ (3, 4) ⑤ (4, 2)

28. Suppose that node 1 and node 7 are not connected.
What is tree.top after the fct ended?

① 6 ② 7 ③ 8 ④ 9 ⑤ 10

29. Suppose that node 1 and node 7 are not connected.

Find the edge that cannot be in the tree.T when tree.top is 3 at Ⓐ.

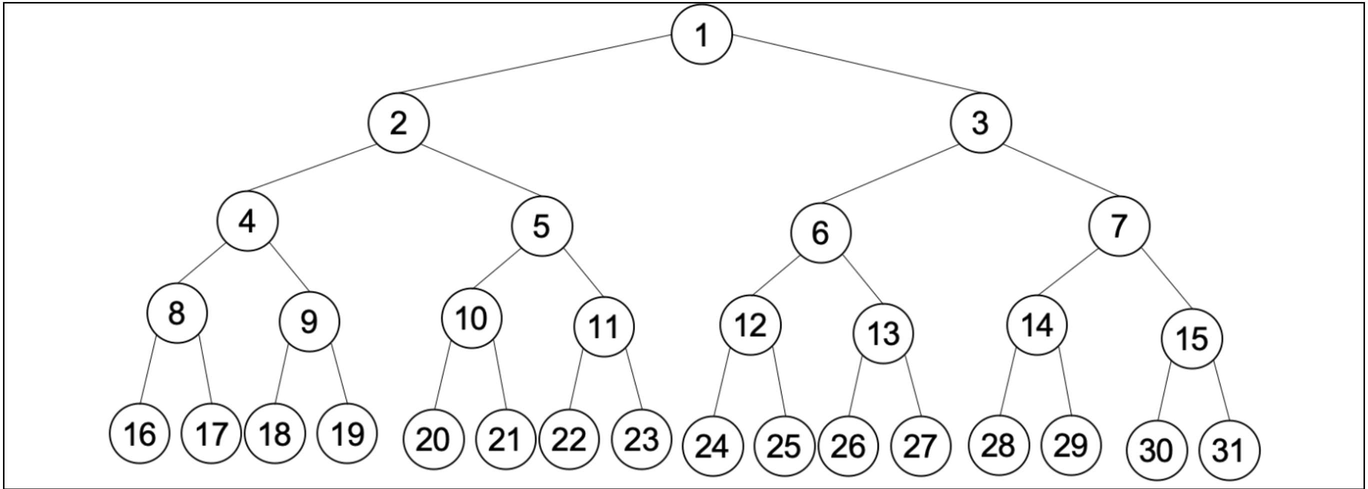
① (0, 1) ② (0, 2) ③ (1, 3) ④ (0, 7) ⑤ (2, 4)

30. Suppose that node 1 and node 7 are not connected.

Find the edge that cannot be in the tree.T when tree.top is 6 at Ⓐ.

① (0, 1) ② (7, 6) ③ (3, 4) ④ (6, 5) ⑤ (1, 4)

Q5. Graph below represent position of the Nodes. Make each Binary search tree, Min heap tree and answer the questions



Binary search tree operations

Insert 17
Insert 5
Insert 24
Insert 15
Insert 30
Insert 34
Insert 2
Insert 1
Insert 19
Insert 22
Insert 28
Insert 37
Insert 11
Insert 13
Insert 4

Max heap tree operations
(Heapify after every insert)

Insert 17
Insert 5
Insert 24
Insert 15
Insert 30
Insert 34
Insert 2
Insert 1
Insert 19
Insert 22
Insert 28
Insert 37

31. [BINARY_TREE]
What is the value of Node 7 after all operations

① NULL ② 34 ③ 13 ④ 30 ⑤ 22

32. [BINARY_TREE]
What is the value of Node 15 after all operations

① NULL ② 34 ③ 13 ④ 30 ⑤ 22

33. [BINARY_TREE]
What is the value of Node 19 after all operations

① NULL ② 34 ③ 13 ④ 30 ⑤ 22

34. [MAX_HEAP]
What is the value of Node 1 after all operations

① NULL ② 28 ③ 34 ④ 30 ⑤ 37

35. [MAX_HEAP]
What is the value of Node 5 after all operations

① NULL ② 24 ③ 19 ④ 30 ⑤ 22

36. [MAX_HEAP]
What is the value of Node 9 after all operations

① NULL ② 5 ③ 15 ④ 22 ⑤ 17