2020 2 학기
자료구조개론
기말고사

1. You will get +1 point for each correct answer, 0 for each unanswered question, and -0.3 for each wrong answer. Be careful when you guess.

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 at the beginning.

Q1. Given static hash with open addressing, bucket size 13 and slot size 1, answer the questions

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

Hash descriptions

- Static hashing
- Bucket size 13
- Slot size 1
- Open addressing
- Linear probe

Hash functions

H1(key)

⇨ **key % 13**

H2(key)

⇨ **(key * key) % 13**

Insert following keys in order

4
2
0
1
44
23
12
5
9
24
32

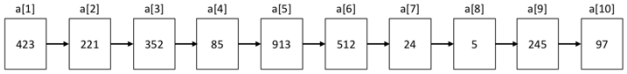
1. After insertion, what is the value at Ⓐ when hash function is H1 (if there's no value write NULL)
2. After insertion, what is the value at Ⓒ when hash function is H1 (if there's no value write NULL)
3. After insertion, what is the value at Ⓓ when hash function is H1 (if there's no value write NULL)
4. After insertion, what is the value at Ⓐ when hash function is H2 (if there's no value write NULL)
5. After insertion, what is the value at Ⓑ when hash function is H2 (if there's no value write NULL)
6. After insertion, what is the value at Ⓓ when hash function is H2 (if there's no value write NULL)

Q2. Read the following program, and answer the questions.

```
int sort(int a[], int link[], int d, int r, int n) {
    int front[r], rear[r];
    int i, bin, current, first, last;

    first = 1;

    for(i = 1; i < n; i++) link[i] = i+1;
    link[n] = 0;

    for(i = d-1; i >=0; i--) {
        for(bin = 0; bin < r; bin++) front[bin] = 0;

        for(current = first; current; current = link[current]){
            bin = digit(a[current], d - i, r);
            if(front[bin] == 0)     front[bin] = current;
            else    link[rear[bin]] = current;
            rear[bin] = current;
        }

        for(bin=0; !front[bin]; bin++);
        first=front[bin]; last=rear[bin];

        (A)

        for(bin++; bin < r; bin++) {
            if(front[bin]) {
                link[last] = front[bin]; last = rear[bin];
            }
        }
        link[last] = 0;

        (B)

    }
    return first;
}
```

```
int digit(int number, int i, int r)
{
    for (int div = 0; div < i; div++)
        number /= r;
    return number % r;
}
```

int a[] = {0, 423, 221, 352, 85, 913, 512, 24, 5, 245, 97};



7. What is the name of this sorting algorithm?

8. What is the time complexity of this algorithm?

9. In calling the function “sort”, what is the proper value of d and r to sort a[]? (<ex> d=-1, r=-1)

10. When i is d-1, what is the value of link[9] at (B)?

11. When i is d-1, what is the value of link[1] at (B)?

12. When i is d-1, what is the value of rear[9] at (B)?

13. When i is d-1, what is the value of the variable “bin” at (A)?

14. When i is d-2, what is the value of link[1] at (B)?

15. When i is 0, what is the value of a[first] at (B)?

16. when i is 0, what is the value of the variable “last” at (B)?

Q3. Read the following graph program, and answer the questions. **Assume "func" is called with <u>v = 0</u>.**

```
void func(int v, int cost[][MAX_VERTICES],
        int distance[], int n, short int found[])
{
      int i, u, w;
      for (i = 0; i < n; i++) {
            found[i] = FALSE;
            distance[i] = cost[v][i];
      }
      found[v] = TRUE;      distance[v] = 0;
      for (i = 0; i < n-2; i++) {
            u = choose(distance, n, found);
            found[u] = TRUE;
            for (w = 0; w < n; w++) {
                  if (!found[w]
                        && distance[u]+cost[u][w] < distance[w])
                        distance[w]= distance[u]+cost[u][w];
            }
      Ⓑ
      }
}
```

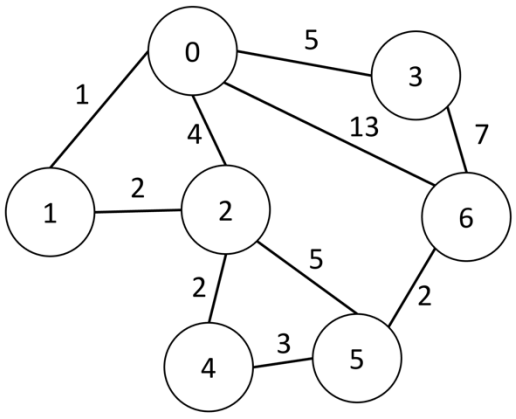
```
int choose(int distance[], int n, short int found[])
{
      /* finds the smallest distance not yet checked */
      int i, min, minpos;
      min = INT_MAX;
      minpos = -1;
      for (i = 0; i < n; i++) {
            if (distance[i] < min && !found[i]) {
                  min = distance[i];
                  minpos = i;
            }
      }
      return minpos;
}
```

MAX_VERTICES 7

```
int cost[][MAX_VERTICES] =
```

0,	1,	4,	5,	1000,	1000,	13
1,	0,	2,	1000,	1000,	1000,	1000
4,	2,	0,	1000,	2,	5,	1000
5,	1000,	1000,	0,	1000,	1000,	7
1000,	1000,	2,	1000,	0,	3,	1000
1000,	1000,	5,	1000,	3,	0,	2
13,	1000,	1000,	7,	1000,	2,	0

/* 1000 implies there is no connection between the vertices. */



17. What is the name of this graph algorithm?

18. When i is 0, what is the value of distance[1] at Ⓑ?

19. When i is 0, what is the value of distance[4] at Ⓑ?

20. When i is 1, what is the value of u at Ⓑ**?**

21. When i is 2, what is the value of u at Ⓑ**?**

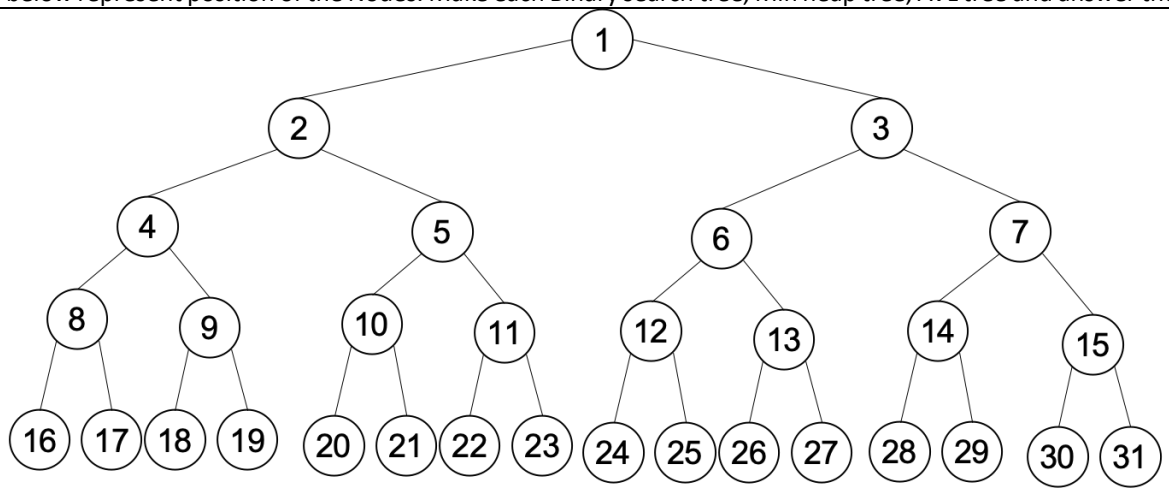
22. When i is 3, what is the value of u at Ⓑ**?**

23. When i is 4, what is the value of distance[5] at Ⓑ**?**

24. If the “func” is called with v = 4, what is the value of distance[5], when i is 0, at Ⓑ

25. If the “func” is called with v = 4, what is the value of distance[3], when i is 1, at Ⓑ**?**

Q4. Graph below represent position of the Nodes. Make each Binary search tree, Min heap tree, AVL tree and answer the questions



Binary search tree operations	Min heap tree operations (Heapify after every insert)
Insert 25 Insert 35 Insert 22 Insert 12 Insert 1 Insert 13 Insert 29 Insert 9 Insert 20 Insert 24 Insert 32 Insert 23 Insert 27	Insert 23 Insert 15 Insert 17 Insert 32 Insert 10 Insert 12 Insert 5 Insert 6 Insert 30 Insert 9 Insert 1

26. [BINARY_TREE]
    What is the value of Node 1 after all operations
    (Write NULL if there's no value)

27. [BINARY_TREE]
    What is the value of Node 5 after all operations
    (Write NULL if there's no value)

28. [BINARY_TREE]
    What is the value of Node 7 after all operations
    (Write NULL if there's no value)

29. [BINARY_TREE]
    What is the value of Node 17 after all operations
    (Write NULL if there's no value)

30. [BINARY_TREE]
    What is the value of Node 18 after all operations
    (Write NULL if there's no value)

31. [MIN_HEAP]
    What is the value of Node 1 after all operations
    (Write NULL if there's no value)

32. [MIN_HEAP]
    What is the value of Node 2 after all operations
    (Write NULL if there's no value)

33. [MIN_HEAP]
    What is the value of Node 6 after all operations
    (Write NULL if there's no value)

34. [MIN_HEAP]
    What is the value of Node 11 after all operations
    (Write NULL if there's no value)

35. [MIN_HEAP]
    What is the value of Node 12 after all operations
    (Write NULL if there's no value)