

2019 자료구조 기말시험

1. (1) 이진트리의 높이를 구하는 함수 `int getHeight(BinaryNode *node)` 를 작성하라. (7점)
(2) 이진트리에서 노드의 개수를 반환하는 함수 `int getCount(BinaryNode *node)`를 작성하라. (7점)

2. (1) 이진탐색트리의 삽입함수 `void insertRecur(BinaryNode *r, BinaryNode *n)`를 작성하라. (7점)

```
void insertRecur(BinaryNode *r, BinaryNode *n) {  
    if(n->getData() == r->getData()) . . .  
    else if(n->getData() < r->getData() . . .  
    else . . .  
}
```

와 같이 구현하라.

- (2) 다음은 이진탐색트리의 삭제함수중 삭제하려는 노드가 2개의 자식이 모두 있는 경우다. 밑줄 친 부분을 구현하라. (10점)

```
else {  
    BinaryNode* succp = node;    // node 는 삭제하려고 하는 노드  
    BinaryNode* succ = node->getRight();  
    while (succ->getLeft() != NULL) {  
        _____  
        _____  
    }  
  
    if( succp->getLeft() == succ )  
        _____  
    else  
        _____  
    node->setData(succ->getData());  
    _____  
}
```

- (3) 20 50 30 40 70 10 17 15를 순차적으로 이진탐색트리에 삽입한 후 root를 삭제한 후의 그림을 그려라. (7점)

3. (1) 10, 40, 30, 5, 12, 6 을 순차적으로 읽어서 최소 힙 트리를 구성하라. (5점)
(2) 최대 heap에서의 삭제 함수이다. 코딩하라. (7점)

```
HeapNode remove() {  
    HeapNode item = node[i];  
    HeapNode last = node[size--];  
    int parent = 1; int child = 2;  
    while(child < size) {  
        ... while 문 내부를 작성할 것  
    }  
    node[parent] = last;  
    return item;  
}
```

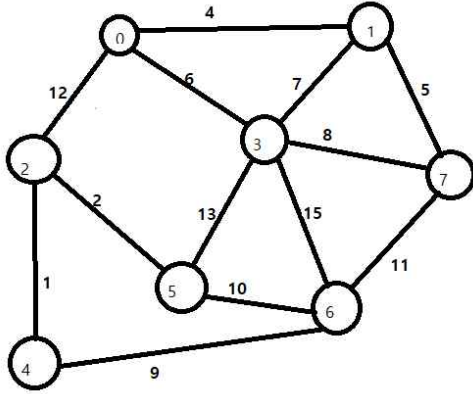
4. 다음 그래프에 대하여

(1) Kruskal 알고리즘을 적용하여 단계적으로 진행해보라.

(7점)

(2) Prim 알고리즘을 적용하여 3번 정점에서 단계적으로 진행해보라.

(7점)



5. 위 6번 그래프에 대하여

(1) 가중치 인접행렬을 그려라.

(4점)

(2) 정점 5를 기준으로 Dijkstra 최단경로 과정을 단계별로 보여라.

(7점)

6. 다음의 정렬기법을 이용하여 아래의 정수 배열을 오름차순으로 정렬하라.

10 8 6 20 4 3 22 1 0 15 16

(1) 쉘 정렬

(7점)

(2) 퀵 정렬

(7점)

(3) 데이터 324 652 685 283 935 365 876 123 에 대하여 기수 정렬 진행과정을 단계별로 보여라.

(7점)

(4) 다음은 머지소트(mergesort)의 구현이다. if 절 안의 4줄을 채워라.

(5점)

```
void mergesort(int list[], int left, int right) {
    int mid;
    if(left < right) {
        // mid 값
        // 왼쪽 부분 리스트 정렬하는 recursive 함수
        // 오른쪽 부분 리스트 정렬하는 recursive 함수
        // 합병 merge() 함수
    }
}
```

7. 데이터 30 40 50 10 20 80 60 45 90 55 47 을 차례대로 삽입한다.

단계별로 AVL 트리의 진행과정을 보여라.

(7점)

8. 다음 해시맵 클래스 내에서 선형조사법을 이용한 삽입함수 addLinearProb 함수를 완성하라. (7점)

```
class HashMap {
    Record table[TABLE_SIZE];
public :
    . . .
    void addLinearProb(char* key, char* value) {
        int i, hashValue;
        // 다음을 완성할 것.
    }
}
```

9. 다음은 Union-Find 연산을 위한 정점 집합 클래스 구현이다. 밑줄친 부분을 채워라. (5점)

```
class VertexSets {
    int parent[MAX_VTXS];
    int nSets;
public:
    VertexSets(int n) : nSets(n) {
        for(int i=0; i<nSets; i++) _____ // 모든 정점이 고유의 집합에 속함
        bool isRoot(int i) { return _____ }
        int findSet(int v) { _____ // v 가 속한 집합을 찾아 반환
            while(!isRoot(v)) _____;
            return v;
        }
        void unionSets(int s1, int s2) {
            _____;
            _____;
        }
    };
};
```