

자료구조론<선택>

2016년 시행 5급 공채(기술) 제2차시험

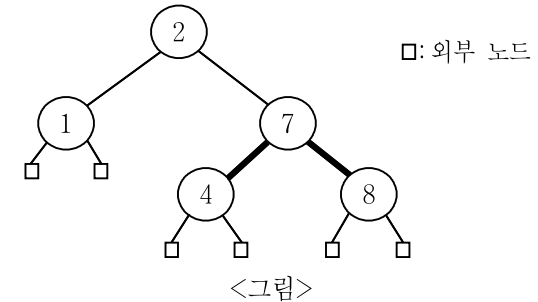
응시번호 :

성명 :

제 1 문. 기수 정렬(radix sorting)에 대한 다음 물음에 답하시오. (총 6점)

- 1) 기수정렬이 다른 정렬 방법(버블 정렬, 퀵 정렬, 힙 정렬 등)들과 근본적으로 다른 가장 큰 특징을 설명하시오. (2점)
- 2) 만일 정렬할 키의 값(itemkey)이 네 자리수(4-digit) 정수일 경우 기수 정렬을 위해 1, 10, 100, 1000 각 자리수의 값을 뽑아내는 효율적인 C 프로그램 표현(expression)을 만들어 보시오. (예를 들어, itemkey가 1234일 경우, 각 자리수의 값 4, 3, 2, 1을 뽑아낼 수 있어야 함) (2점)
- 3) N개의 10진수를 정렬하라. 이 때 10진수의 자리수를 P라고 할 때, 기수 정렬의 시간 복잡도를 big-O로 표현하고 근거를 설명하시오. (2점)

제 2 문. 레드-블랙 트리(red-black tree)는 균형 트리(balanced tree)의 일종으로 루트에서 외부 노드에 이르는 경로 상에 두 개의 연속된 레드 링크가 나오지 않으며, 루트에서 각 외부 노드까지의 경로에 있는 블랙 링크의 수가 모두 같다는 성질을 가지고 있다. 레드-블랙 트리에서 루트 노드와 외부 노드는 모두 블랙 노드이다. 레드-블랙 트리에서 가는 실선은 블랙 링크를 표시하며 블랙 링크에 연결된 자식 노드는 블랙 노드임을 나타내고, 굵은 실선은 레드 링크를 표시하며 레드 링크에 연결된 자식 노드는 레드 노드임을 가정한다. 다음 <그림>은 공백 트리에 키 값으로 2, 1, 7, 8, 4를 가지는 노드를 순서대로 삽입했을 경우 생성되는 레드-블랙 트리의 예를 보여주고 있다.



레드-블랙 트리에 관한 다음 물음에 답하시오. (총 13점)

- 1) 루트에서 외부 노드에 이르는 경로 상에 레드 링크가 연속으로 나올 경우 트리는 불균형(imbalance)이 되었다고 하며, 이러한 경우 노드를 회전시켜 주어야 한다. 회전에는 단일 회전(single rotation)과 이중 회전(double rotation)이 있는데, 단일 회전이 필요한 2가지 경우와 이중 회전이 필요한 2가지 경우를 나열하고, 각각의 경우에 대해 회전이 수행되는 과정을 그림을 그려서 설명하시오. (6점)
- 2) <그림>의 레드-블랙 트리에 키 값 6과 5를 가지는 노드를 연속해서 삽입한다. 이 때 키 값 6을 가지는 노드를 삽입했을 때 트리가 변화하는 과정과 계속해서 키 값 5를 가지는 노드를 삽입했을 때 트리가 변화하는 과정을 그림을 그려서 설명하시오. (7점)

제 3 문. 다음은 완전이진트리(complete binary tree) 형태로 저장된 정수형 자료를 입력으로 받아 최대힙(max heap)을 생성하는 알고리즘 makeMaxHeap의 의사코드이다.

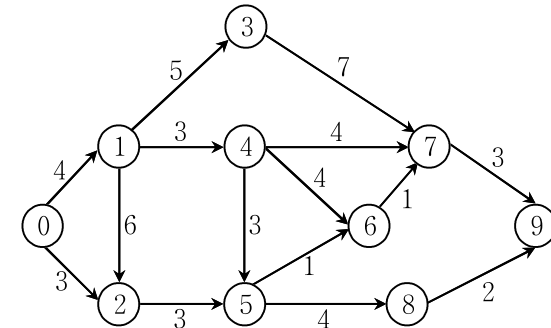
```
void makeMaxHeap(H) {
    if (H is not a leaf) {
        makeMaxHeap(left subtree of H);
        makeMaxHeap(right subtree of H);
        K = root(H);
        maxHeapify(H,K);
    }
    return;
}
```

위 알고리즘에서 root(H)는 트리 H의 루트노드에 저장된 값을 반환하는 함수이며, maxHeapify(H,K)는 H의 왼쪽 부분트리와 오른쪽 부분트리가 최대힙일 때, 저장된 값이 K인 루트노드를 포함한 H 전체가 최대힙이 되도록 만드는 함수이다. 다음 물음에 답하시오. (총 12점)

- 1) H에 저장된 자료의 수가 n일 때, makeMaxHeap의 최악수행시간을 분석하시오. (7점)
- 2) n개의 정수가 저장된 임의의 배열 A[1:n]이 완전이진트리 구조를 만족하는 경우 알고리즘 makeMaxHeap에 의해 최대힙으로 변형될 수 있다. 처음 A의 상태가 다음과 같을 때, makeMaxHeap에 의해 최대힙으로 변경된 배열 A의 결과를 보이시오. (5점)

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A[i]	8	3	16	7	1	9	4	5	17	11	15	6	10	2	13

제 4 문. 다음은 어떤 프로젝트에서 수행되어야 할 작업들과 이에 따른 사건을 그래프로 모델링한 것이다. 간선은 프로젝트의 작업을 표시한다. 정점으로 표시되는 각 사건은 그에게 들어오는 모든 작업이 완료될 때 일어나고, 다음 작업이 시작될 수 있다는 것을 의미한다. 예를 들어, 사건 5(정점 5)는 간선 (2, 5)와 간선 (4, 5)에 의해 표현되는 작업이 모두 종료됨을 의미하고 동시에 간선 (5, 6)과 간선 (5, 8)에 의해 표현되는 작업을 시작할 수 있다는 것을 의미한다. 정점 0과 정점 9는 프로젝트의 시작과 종료로 해석될 수 있다. 간선에 부여된 숫자는 작업 수행에 필요한 시간이고, 각 작업은 병렬적으로 수행될 수 있다. 다음 물음에 답하시오. (총 10점)



- 1) 모든 사건 i ($0 \leq i \leq 9$)에 대해서 그 사건이 일어날 수 있는 가장 이른 시간을 구하시오. (사건 0의 시작시간은 0이라 가정한다) (2점)
- 2) 개별 작업 시간을 단축하면 전체 작업의 소요 시간을 단축시킬 수 있는 개별 작업이 있는가? 있다면 그런 작업을 나열하고 이유를 설명하시오. (3점)
- 3) 정점의 개수를 V, 간선의 개수를 E라 했을 때 1)을 효율적으로 계산하는 알고리즘을 제시하고 제시한 알고리즘에 대한 시간복잡도를 설명하시오. (5점)

제 5 문. {E, G, I, L, N, T}의 6개로 이루어진 문자로만 총 70자의 정보를 전송하려고 한다. 일반적으로는 고정크기의 비트 수를 할당하여 정보를 보내며, 받은 정보는 같은 코드로 해석한다. 만약에 각 문자의 빈도 수가 다를 경우 허프만(Huffman) 트리를 이용하여 구한 허프만 코드를 활용하면 정보전송 효율을 높일 수 있다. 허프만 트리에서 왼쪽 간선은 0, 오른쪽 간선은 1로 가정한다. 각 문자의 빈도 수가 다음과 같을 때 물음에 답하시오. (총 9점)

{E: 25, G: 5, I: 10, L: 3, N: 12, T: 15}

- 1) 각 문자에 대한 허프만 코드를 구하고, 고정크기 코드와 허프만 코드를 사용하여 모든 정보를 전송하는데 사용되는 비트 수를 각각 구하시오. (단, 고정크기 코드는 3 비트이다) (5점)
- 2) {INTELLIGENT}란 문자열을 1)에서 구한 허프만 코드를 사용하여 비트 문자열로 표시하시오. (2점)
- 3) {001010011011111001000111001}의 비트 문자열을 1)에서 구한 허프만 코드를 이용하여 문자열로 변경하시오. (2점)