

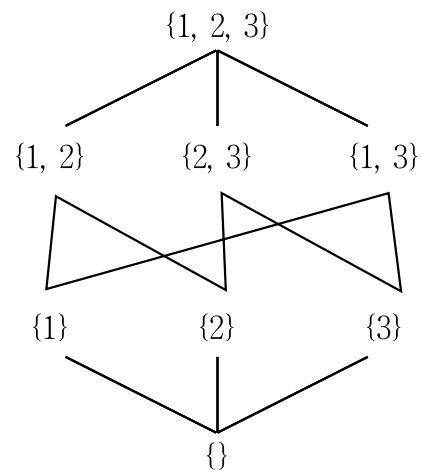
자료구조론<선택>

2019년도 국가공무원 5급(기술) 공개경쟁채용 제2차시험

응시번호 :

성명 :

제 1 문. 원소 개수가 n 개인 집합의 멱집합(powerset)은 격자구조(lattice)로 표현이 가능하다. 격자구조에서 각 정점은 각 부분집합이며, 두 부분집합 A, B 가 $A \subset B$ 이고, $A \subset C \subset B$ 인 집합 C 가 존재하지 않을 경우 두 정점 간에 간선이 존재한다. 예를 들어 집합 $\{1, 2, 3\}$ 에 대한 멱집합을 격자구조로 표현하면 다음과 같다. 물음에 답하시오. (총 10점)



- 1) 원소 개수가 n 개인 집합에서 원소 개수가 k 개인 모든 부분집합의 개수를 구하기 위한 `int numSubsets(n, k)` 함수에 대해 C언어 형식의 의사코드를 제시하고, 제시한 의사코드의 시간복잡도를 빅오(Big-Oh) 표기법으로 표현하시오. (단, 가능한 한 효율적인 구현을 제시한다) (5점)

- 2) 원소 개수가 n 개인 집합의 멱집합을 표현하는 격자구조가 있다고 하자. 이때 해당 격자구조에 존재하는 모든 간선의 개수를 구하는 `int numEdges(n)` 함수에 대해 C언어 형식의 의사코드를 제시하고, 제시한 의사코드의 시간복잡도를 빅오(Big-Oh) 표기법으로 표현하시오. (단, 가능한 한 효율적인 구현을 제시한다) (5점)

제 2 문. 크기가 13인 해시 테이블에서 다음과 같은 해시 함수가 주어지고 10개의 데이터를 차례대로 삽입한다고 할 때, 물음에 답하시오. (총 10점)

- 해시 함수: $h(k) = k \bmod 13$
- 데이터: 38, 14, 10, 12, 3, 26, 8, 7, 5, 18

- 1) 오버플로우(overflow) 해결을 위해 선형조사법(linear probing)을 사용하는 경우 해시 테이블을 보이시오. (2점)
- 2) 오버플로우 해결을 위해 이중 해싱(double hashing) 기법을 사용하는 경우 해시 테이블을 보이시오. (단, 두 번째 해시 함수 $h'(k)$ 는 다음과 같다) (3점)

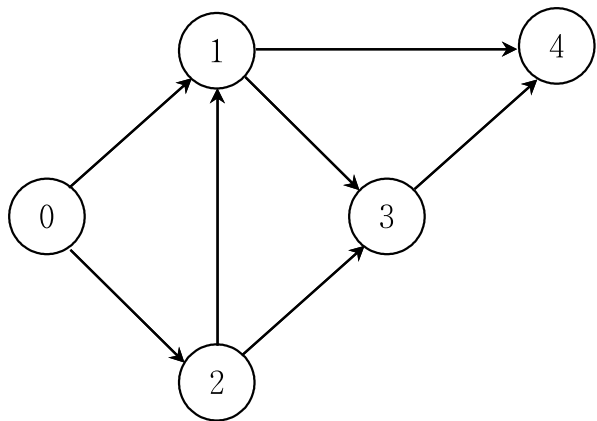
$$h'(k) = 5 - (k \bmod 5)$$

- 3) 오버플로우 해결을 위해 이차조사법(quadratic probing)을 사용하는 경우 해시 테이블을 보이시오. (단, 다음 조사할 위치를 결정하는 식 $h'(k)$ 는 다음과 같다) (2점)

$$h'(k) = (h(k) + \text{inc}^2) \bmod 13 \text{ for } \text{inc} = 1, 2, \dots$$

- 4) 1), 2), 3)의 수행결과를 사용하여 선형조사법, 이중 해싱, 이차조사법의 성능을 충돌 횟수 기준으로 비교하시오. (3점)

제 3 문. 다음은 어떤 프로젝트에서 수행해야 할 작업 간의 선행관계를 나타내는 AOV (activity on vertex) 네트워크이다. 이때 정점은 작업(activity)을 나타내며 방향 간선은 작업 간의 선행관계를 나타낸다. 물음에 답하시오. (총 10점)



1) 다음의 선행관계에 대한 정의 중에서 수행 가능한 프로젝트를 표현하는 AOV 네트워크에 존재하지 않아야 하는 것을 모두 고르시오. (2점)

- ㉠ 이행적(transitive) 선행관계
- ㉡ 반사적(reflexive) 선행관계
- ㉢ 부분 순서(partial order)
- ㉣ 방향 사이클
- ㉤ 위상 순서(topological order)

2) 다음은 주어진 AOV 네트워크가 수행 가능한 프로젝트인지 판단하는 알고리즘의 C언어 형식의 의사코드이다. 입력 매개변수인 그래프(graph)는 주어진 AOV 네트워크를 의미하고, n은 총 정점의 수를 의미한다. 수행 가능한 프로젝트일 경우 true, 수행이 불가능한 프로젝트인 경우 false를 반환하여야 할 때, 의사코드가 올바르게 수행되기 위해 빈칸 ㉠과 ㉡을 완성하시오. (3점)

01: typedef struct node *nPointer;	20: for (i=0; i<n; i++) {
02: typedef struct {	21: if (t == -1)
03: int vertex;	22: return false;
04: nPointer link;	23: else {
05: } node;	24: j = t;
06: typedef struct {	25: t = ㉠
07: int count;	26: for (ptr = graph[j].link; ptr;
08: nPointer link;	27: ptr = ptr->link) {
09: } heads;	28: k = ㉡
10:	29: graph[k].count--;
11: bool isWorkable(heads graph[], int n) {	30: if (graph[k].count == 0) {
12: int i, j, k, t=-1;	31: graph[k].count = t;
13: nPointer ptr;	32: t = k;
14: for (i=0; i<n; i++) {	33: }
15: if (graph[i].count == 0) {	34: }
16: graph[i].count = t;	35: }
17: t = i;	36: return true;
18: }	37: }
19: }	

3) 주어진 AOV 네트워크에 대해 2)의 의사코드를 실행할 경우, 20번째 줄의 for 반복문을 반복할 때마다 24번째 줄의 j의 값이 어떻게 변경되는지를 모든 i에 대하여 각각 보이시오. (2점)

4) 주어진 AOV 네트워크의 총 정점의 개수를 n, 총 간선의 개수를 e라고 할 때에, 2)의 의사코드의 시간복잡도를 빅오(Big-Oh) 표기법으로 표현하고 이를 증명하시오. (3점)

제 4 문. 다음과 같은 순서로 데이터를 탐색 구조에 삽입할 때, 물음에 답하시오.
(총 10점)

7, 5, 11, 10, 2, 3, 6, 8, 15, 13

- 1) 이진 탐색 트리(binary search tree), 최소 힙(min heap), AVL 트리, 레드블랙 트리(red-black tree)의 결과 트리를 그리시오. (단, 레드블랙 트리에서 블랙 링크는 단일 실선(−), 레드 링크는 이중 실선(=)으로 표시 한다) (8점)
- 2) n개의 원소가 삽입되어 있는 이진 탐색 트리, 최소 힙, AVL 트리, 레드블랙 트리에 새로운 원소를 삽입할 때, 최악의 경우의 시간복잡도를 빅오(Big-Oh) 표기법으로 표현하시오. (2점)

제 5 문. 선택 트리(selection tree)는 완전 이진 트리(complete binary tree)를 사용하여 토너먼트 경기 방식을 통해 트리를 구축하는 자료구조로서 승자 트리(winner tree)와 패자 트리(loser tree)가 있다. 선택 트리를 사용하여 키 값이 가장 작은 원소를 선택한다고 할 때, 승자 트리에서는 형제 노드끼리 값을 비교하여 값이 작은 원소가 승자가 되어 부모 노드로 복사된 다음 토너먼트 경쟁을 계속하는 반면, 패자 트리에서는 키 값이 큰 패자는 부모 노드에 복사되고, 키 값이 작은 승자는 그 위 부모 노드로 올라가서 다시 토너먼트 경쟁을 계속하는 형태로 트리를 구성한다. 따라서 승자 트리에서는 루트(1번 노드)에 결승 토너먼트의 승자가, 패자 트리에서는 루트(1번 노드)에 결승 토너먼트의 패자가 나타나며, 전체 승자는 루트 위에 별도로 위치한 0번 노드에 나타난다. 다음과 같이 오름차순으로 정렬된 8개의 런(run)이 있을 때, 물음에 답하시오. (단, 각 런의 첫 번째 원소를 완전 이진 트리의 단말(leaf) 노드로 복사한 다음 선택 트리의 구축이 시작된다)
(총 10점)

런1: 8, 11
런2: 13, 14
런3: 2, 16, 21
런4: 5, 17
런5: 9, 19, 25
런6: 4, 15, 22
런7: 7, 10
런8: 1, 12, 20

- 1) 위의 런으로부터 만들어지는 승자 트리를 그리시오. (2점)
- 2) 위의 런으로부터 만들어지는 패자 트리를 그리시오. (2점)
- 3) 1)에서 만들어진 선택 트리로부터 가장 작은 원소 2개를 삭제한다고 할 때, 원소를 삭제할 때마다 변경되는 트리를 순서대로 그리시오. (3점)
- 4) 2)에서 만들어진 선택 트리로부터 가장 작은 원소 2개를 삭제한다고 할 때, 원소를 삭제할 때마다 변경되는 트리를 순서대로 그리시오. (3점)

인사혁신처 시험출제과장