

2015 학년도 2 학기 기말고사 (1/2)	과목명	자료구조 (분반: 00□)	학 과	학 년	점 수
담당교수	심 정 섭		학 번		
시험일시	12월 14일 월요일		성 명		

1. 아래 의사코드(pseudo code)는 우선순위큐 추상자료형(priority queue ADT)를 이용하여 데이터를 정렬하는 알고리즘이다. 입력시퀀스  $S$ 가 크기  $n$ 인 배열로 주어졌을 때, (1)우선순위큐를 구현하는 세 가지 방법(자료구조)과, 이를 이용하여 아래  $PQ-Sort$  알고리즘을 구현하였을 때, 각 구현 방법에 대해 (2)첫 번째 while문을 수행하는 최악수행시간, (3)두 번째 while문을 수행하는 최악수행시간을 강의 시간에 설명한 대로 아래 표에 쓰시오. (단, 각 방법에 대해 구현 가능한 가장 효율적인 시간을 쓰시오.)

**Algorithm  $PQ-Sort(S, C)$**

**Input** sequence  $S$ , comparator  $C$  for the elements of  $S$

**Output** sequence  $S$  sorted in increasing order according to  $C$

$P \leftarrow$  priority queue with comparator  $C$

**while**  $\neg S.empty()$

$e \leftarrow S.front(); S.eraseFront()$

$P.insert(e, \emptyset)$

**while**  $\neg P.empty()$

$e \leftarrow P.removeMin()$

$S.insertBack(e)$

(1)구현방법	(2)첫 번째 while문의 최악수행시간	(3)두 번째 while문의 최악수행시간
	$O(\quad)$	$O(\quad)$
	$O(\quad)$	$O(\quad)$
	$O(\quad)$	$O(\quad)$

2. 강의시간에 설명한 사전(dictionary) ADT를 구현하는 방법(명칭)들 중 4개를 쓰고, 각 방법들에 대해 다음 연산들의 입력크기  $n$ 에 대한 최악수행시간을 쓰시오.

명칭	$find(k)$	$put(k,o)$	$erase(k)$
	$O(\quad)$	$O(\quad)$	$O(\quad)$
	$O(\quad)$	$O(\quad)$	$O(\quad)$
	$O(\quad)$	$O(\quad)$	$O(\quad)$
	$O(\quad)$	$O(\quad)$	$O(\quad)$

3. 크기 17인 배열  $A$ 와, 다음과 같은 1차( $h_1$ ) 및 2차( $h_2$ ) 해시함수를 이용하여 이중해싱(double hashing)을 구현하였다. 다음 물음에 답하시오.

$$\begin{cases} h_1(k) = k \bmod 17 \\ h_2(k) = 7 - (k \bmod 7) \end{cases}$$

(1) 해시테이블에 다음 값들이 각각 키(key)인 자료들이 차례로 삽입되었을 때, 배열  $A$ 의 마지막 상태를 빈칸을 모두 채워 표시하시오. 단, 처음에 테이블의 각 원소는 모두 0으로 초기화되어 있다.

입력: 8, 41, 30, 24, 39, 55, 22, 31, 73, 43, 9

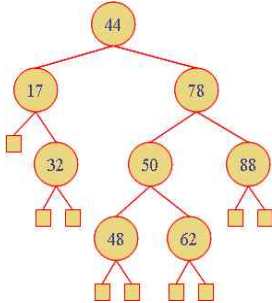
$i$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$A[i]$																	

(2) 강의시간에 설명한 방법으로, (1)의 과정이 수행되는 동안의 총 탐사(probe) 수를 쓰시오.

답: (                    )회

(계산용 여백)

4. 아래 그림은 키가 44인 노드가 루트인 AVL 트리이다. 강의 시간에 설명한 방법을 이용하여 트리에 키가 47인 자료를 삽입했을 때, 결과 AVL 트리를 그림으로 나타내시오.



5. 다음은  $n$ 개의 정점(vertex)과  $m$ 개의 간선(edge)으로 구성된 그래프에 대한 두 가지 구현 방법을 비교한 표이다. 빈 칸들을  $O$ -표기법으로 나타내시오. 단,  $O$ 는 생략한다.

	인접리스트 (adjacency list)	인접행렬 (adjacency matrix)
space	①	②
$v.incidentEdges()$	③	④
$v.isAdjacentTo(w)$	⑤	⑥
$insertVertex(o)$	⑦	⑧
$eraseVertex(v)$	⑨	⑩

6. 아래 의사코드는 입력그래프  $G$ 에 대한 위상정렬(topological sorting)을 수행하는 알고리즘이다.

**Algorithm TopologicalSort( $G$ )**

$H \leftarrow G$

$n \leftarrow G.numVertices()$

**while**  $H$  is not empty **do**

    (1)

    Label  $v \leftarrow n$

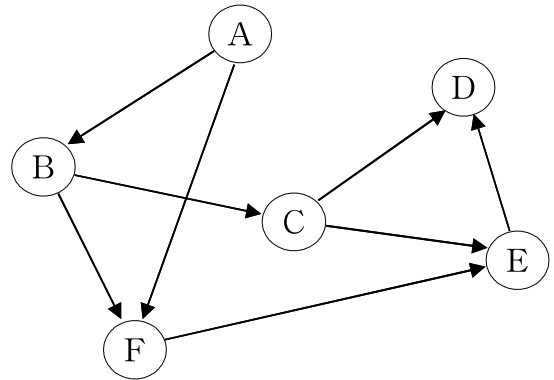
$n \leftarrow n - 1$

    Remove  $v$  from  $H$

(1) 빈 칸에 적절한 코드를 쓰시오.

답:

(2) 알고리즘 TopologicalSort를 아래 그래프에 적용시켰을 때, 결과를 그래프에 표시하시오.



(3) 위의 그래프에서 유향간선(directed edge)을 모두 무향간선(undirected edge)으로 교체한 후  $A$ 에서 깊이 우선탐색(DFS)과 너비우선탐색(BFS)을 각각 적용하였다. 이때, “discovery edge”로 처리되는 간선들을 순서대로 쓰시오. 단, 한 정점에 인접한 정점이 2개 이상 있을 때, 정점 기호가 사전순서로 빠른 정점을 먼저 방문하기로 한다.

① DFS:

② BFS:

(계산용 여백)