

2023학년도 2학기 자료구조 기말고사

2023. 12. 18

점수

학번:

성명:

※ (1 - 10) 아래 문제를 읽고 물음에 답하십시오. [50점 각 5점]

1. 어떤 트리 3개에 대한 모든 노드 개수의 합이 100이면 간선(edge) 개수의 합은?

- ① 94
- ② 95
- ③ 96
- ④ 97

2. 어떤 이진 트리의 전위 순회(preorder traversal)와 중위 순회(inorder traversal)한 순서가 다음과 같을 때, 이 트리의 단말 노드만을 모두 나열한 것은?

- 전위 순회: A, B, D, E, C, F, G, H, I
- 중위 순회: E, D, B, A, G, F, H, C, I

- ① A, D, E
- ② B, C, D, F
- ③ D, E, F, I
- ④ E, G, H, I

3. 배열(array)로 표현한 이진 트리 중에서 최소 힙(minimum heap)에 해당하지 않는 것은?

- ①

1	3	13	7	9	10	20	17	25
---	---	----	---	---	----	----	----	----
- ②

1	3	13	9	7	20	17	25	10
---	---	----	---	---	----	----	----	----
- ③

1	3	17	7	13	25	20	9	10
---	---	----	---	----	----	----	---	----
- ④

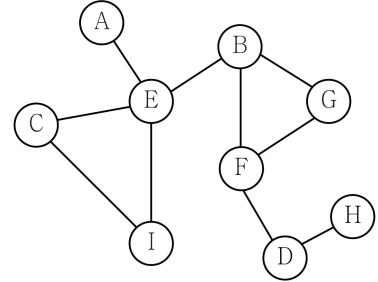
1	9	3	17	10	13	7	20	25
---	---	---	----	----	----	---	----	----

4. 루트 노드의 인덱스를 1로 하는 1차원 배열을 이용하여 최대 힙(max heap)을 구현한 후, 다음 데이터를 차례대로 하나씩 힙에 삽입하였다. 이후 힙 삭제 연산을 1회 수행한 후 배열의 인덱스 6에 저장된 데이터는?

- | |
|------------------------------|
| 15, 8, 10, 18, 22, 13, 26, 7 |
|------------------------------|
- ① 8
 - ② 10
 - ③ 13
 - ④ 15

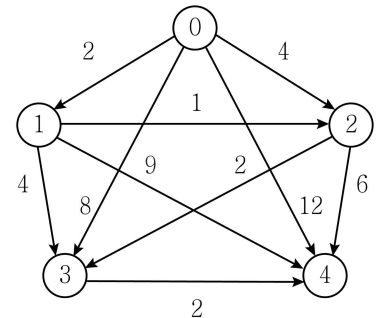
5. 다음 그래프(graph)에 대하여 깊이 우선 탐색(DFS, depth first

search)을 하려고 한다. 시작 정점(vertex)이 B일 때, 정점 I는 몇 번째에 방문하는가? (단, B가 첫 번째 방문 정점이고, 탐색 시 인접한 정점들은 알파벳 오름차순으로 방문한다)



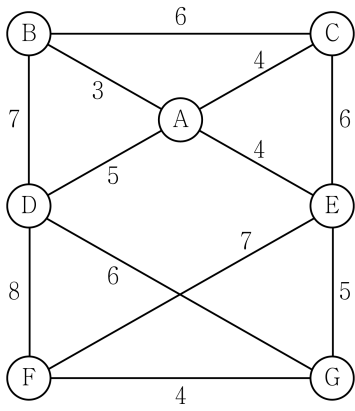
- ① 4
- ② 5
- ③ 8
- ④ 9

6. 다음은 정점 간의 거리를 표시한 방향 그래프(directed graph)이다. 시작 정점이 0일 때, 정점 3과 정점 4까지 최단 경로의 거리를 바르게 연결한 것은?



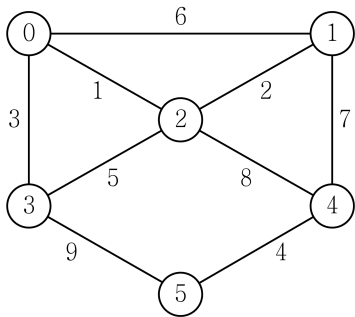
- | | 정점 3 | 정점 4 |
|---|------|------|
| ① | 5 | 7 |
| ② | 5 | 8 |
| ③ | 8 | 11 |
| ④ | 8 | 12 |

7. 다음 그래프에서 크루스칼 알고리즘(Kruskal algorithm)을 사용하여 만든 최소 비용 신장 트리(minimum cost spanning tree)에 대한 모든 간선의 가중치 합은?



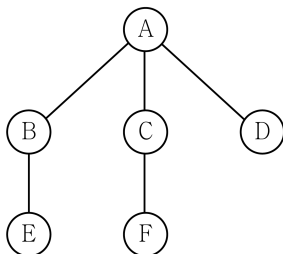
- ① 21
- ② 25
- ③ 31
- ④ 33

8. 다음 그래프에서 프림 알고리즘(Prim algorithm)을 사용하여 최소 비용 신장 트리를 만들 때, 트리에 네 번째로 포함되는 간선은? (단, 시작 정점은 0이다)



- ① (0, 1)
- ② (1, 4)
- ③ (2, 3)
- ④ (4, 5)

9. 다음 그래프에서 정점 A부터 너비 우선 탐색(BFS, breadth first search)을 사용하여 나올 수 있는 결과로 옳지 않은 것은?



- ① A B C D E F
- ② A C B D F E
- ③ A C F B E D
- ④ A D C B F E

20. 다음 인접 행렬(adjacency matrix)로 표현된 그래프에 대해

위상 정렬(topological sort)을 수행한 후, 위상 순서(topological order)에서 마지막에 위치하는 정점은? (단, 위상 정렬 과정에서 선택 가능한 정점이 여러 개인 경우 가장 작은 숫자의 정점을 선택한다)

	0	1	2	3	4	5	6	7	← 도착 정점
0		1		1			1		
1			1						
2									
3		1							
4	1		1			1			
5							1		
6				1				1	
7				1					

↑ 시작 정점

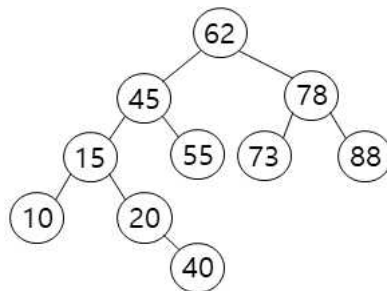
- ① 1
- ② 2
- ③ 3
- ④ 6

11. 탐색을 효율적으로 하기 위한 다양한 탐색 트리에 관하여 다음의 물음에 답하시오. (30점)

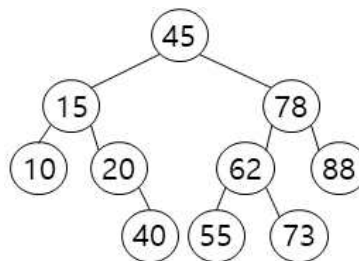
- (1) 다음의 원소를 차례로 삽입하였을 때 다음 각 트리의 구성 결과를 그리시오. (8점)

62, 45, 15, 20, 78, 88, 73, 55, 10, 40

- 1) 이진 탐색 트리 (Binary Search Tree)

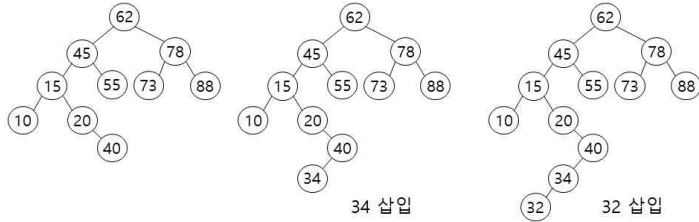


- 2) AVL 트리 (Adelson-Velskii and Landis Tree)

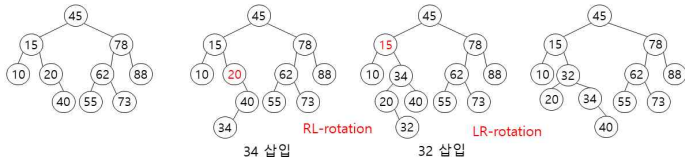


(2) (1)의 결과에 34와 32를 차례로 삽입하는 과정을 상세히 기술하시오. (8점)

1) 이진 탐색 트리



2) AVL 트리



(3) (2)의 완성된 트리로부터 32와 73을 각각 찾기 위해 발생 한 키와 원소들 간의 총 비교 횟수와 트리 노드의 총 방문횟수를 구하시오. (단, 노드에 값이 여러 개인 경우에는 작은 값부터 비교한다.) (5점)

1) 이진 탐색 트리

비교횟수: 7번(32 탐색할 때) 3번(73 탐색할 때)

방문횟수: 7번(32 탐색할 때) 3번(73 탐색할 때)

2) AVL 트리

비교횟수: 3번(32 탐색할 때) 4번(73 탐색할 때)

방문횟수: 3번(32 탐색할 때) 4번(73 탐색할 때)

(4) (2), (3)의 결과에 따라 두 가지 트리의 삽입 및 탐색 성능을 논하고, 각 트리가 최적의 성능을 보이는 경우를 설명하시오. (9점)

- BST와 AVL 트리에서의 삽입 및 탐색 연산은 공통적으로 루트부터 탐색을 시작하여 최악의 경우에 leaf node까지 내려가고, 삽입 연산은 다시 루트까지 거슬러 올라가야 함
- 따라서 BST와 AVL 트리의 삽입 및 탐색 성능 오직 트리 높이 h에 의해 결정됨
- BST는 트리가 한쪽으로 치우쳐 자라나는(skewed tree) 최악의 경우엔 높이가 $h = n$ 이 되어 삽입 및 탐색 연산이 매우 느려 짐
- 반면 AVL 트리는 삽입, 삭제 연산을 반복하더라도 n개의 노드를 갖는 BST의 높이를 항상 $h = \log n$ 이 되도록 유지 가능한, 즉 트리가 한쪽으로 치우쳐 자라나는 현상을 방지하여 트리 높이의 균형(balance)을 유지함

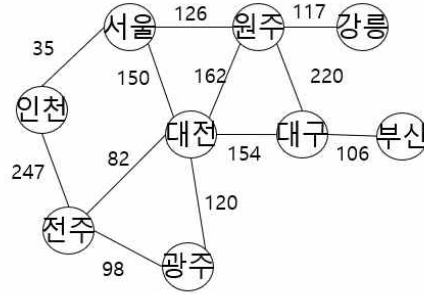
12. 다음은 9개 도시 간의 거리를 나타내고 있다. (20점)

울-대전: 150, 원주-대구: 220, 대전-원주: 162, 광주-대전: 120 (단, 단위는 km)

각 도시와 거리에 대한 자료에 관하여 다음의 물음에 답하시오.

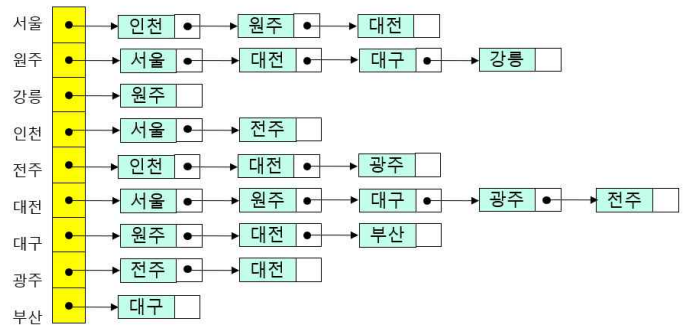
(1) 가중치 그래프로 표현하시오. (8점)

12-(1) 가중치 그래프(4점)



(2) 인접 리스트와 인접행렬로 작성하시오. (단, 인접리스트의 가중치는 무시하고, 인접행렬의 가중치는 포함한다.) (8점)

12-(2) 인접리스트(4점)



또는



서울-인천: 35, 부산-대구: 106, 광주-전주: 98, 강릉-원주: 117, 대전-대구: 154, 전주-대전: 82, 인천-전주: 247, 서울-원주: 126, 서

12-(2) 인접행렬(4점)

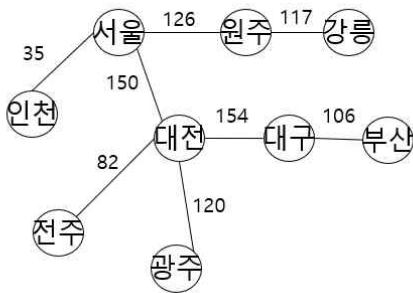
	서울	원주	강릉	인천	전주	대전	대구	광주	부산
서울	0	126	0	35	0	150	0	0	0
원주	126	0	117	0	0	162	220	0	0
강릉	0	117	0	0	0	0	0	0	0
인천	35	0	0	0	247	0	0	0	0
전주	0	0	0	247	0	82	0	98	0
대전	150	162	0	0	82	0	154	120	0
대구	0	220	0	0	0	154	0	0	106
광주	0	0	0	0	98	120	0	0	0
부산	0	0	0	0	0	0	106	0	0

또는

	서울	원주	강릉	인천	전주	대전	대구	광주	부산
서울	0	126	0	35	0	150	0	0	0
원주	126	0	117	0	0	162	220	0	0
강릉	0	117	0	0	0	0	0	0	0
인천	35	0	0	0	247	0	0	0	0
전주	0	0	0	247	0	82	0	98	0
대전	150	162	0	0	82	0	154	120	0
대구	0	220	0	0	0	154	0	0	106
광주	0	0	0	0	98	120	0	0	0
부산	0	0	0	0	0	0	106	0	0

- (3) 서울에서 시작하는 최소거리 신장 트리를 완성하여 보여주고, 선택되는 도시의 순서와 그 도시까지의 거리를 각각 산출하십시오. (8점)

12-(3) 최소거리 신장 트리(4점)



12-(3) 선택되는 도시의 순서와 거리(4점)

순서	도시명	거리
1	인천	35
2	원주	126
3	대전	150
4	전주	232
5	강릉	243
6	광주	270
7	대구	304
8	부산	376