

알고리즘 Quiz 2 답안

충남대학교 컴퓨터공학과
알고리즘 04 분반
201701975 학번 구건모

#문제 1 번

```
public boolean edgeDoesExist(WeightedEdge anEdge){
    int tailVertex = anEdge.tailVertex();
    /* anEdge의 tailVertex를 getter로 받아옴 */
    int headVertex = anEdge.headVertex();
    /* anEdge의 headVertex를 getter로 받아옴 */
    if (this.edgeIsValid(tailVertex, headVertex){
        /* 유효한 Edge 인지 edgeIsValid 메서드로 검사 */
        return (this.adjacency()[tailVertex][headVertex] != WeightedGraph.INFINITE);
        /* 그래프에서 Edge가 존재하지 않을 때 INFINITE 값을 갖는다고 정의하였으므로 Matrix에서
        anEdge에 해당하는 값이 INFINITE가 아니면 Edge가 존재함을 의미하므로 true를 반환,
        INFINITE이면 False 반환 */
    }
    return false; /* edge가 유효하지 않을 경우 false를 반환 */
}
```

#문제 2 번

```
public boolean addEdge(WeightedEdge anEdge){
    if (this.edgeIsValid(anEdge) && !this.edgeDoesExist(anEdge){
        int tailVertex = anEdge.tailVertex();
        /* 그래프에 추가할 Edge의 tailVertex의 WeightedEdge에 구현되어있는
        getter(anEdge.tailVertex())로 얻어옴 */
        int headVertex = anEdge.headVertex();
        /* 그래프에 추가할 Edge의 headVertex의 WeightedEdge에 구현되어있는
        getter(anEdge.headVertex())로 얻어옴 */
        this.adjacency()[tailVertex][headVertex] = anEdge.weight();
        /* 그래프에 anEdge의 weight를 넣어 Edge 추가 */
        this.adjacency()[headVertex][tailVertex] = anEdge.weight();
        /* undirected graph 이므로 역방향도 추가 */
        this.setNumberOfEdges(this.numberOfEdges()+1);
        /* graph에 edge가 추가되었으므로 numberOfEdges를 하나 늘려줌 */
        return true; /* 정상적으로 추가가 완료되었으므로 true를 리턴. */
    }
    return false; /* Edge가 유효하지 않거나, 이미 그래프에 존재하는 Edge일 경우,
    false를 반환 */
}
```

#문제 3번

단계	MinPriorityQ 로 부터 차례대로 연어시는 edge		Tree Edge 여부 판단 (add / discard)	현 단계까지 찾아진 트리의 모습
	Edge	Weight		
초기화	-	-	-	0 1 2 3 4 5
1	(2,4)	2	Add	
2	(0,2)	5	Add	
3	(1,4)	7	Add	
4	(4,5)	8	Add	
5	(0,1)	9	Discard	
6	(1,2)	10	Discard	
7	(2,3)	12	Add	

1. 최초에 Lowest Cost를 가진 Edge인 (2,4)를 Add 합니다.
 2. 이제 (0,2)가 Lowest Cost를 가진 Edge이고 추가했을 때 cycle을 형성하지 않으므로 Add 합니다.
 3. 이제 (1,4)가 Lowest Cost Edge 이고 추가했을 때 Cycle을 형성하지 않으므로 Add 합니다.
 4. 이제 (4,5)가 Lowest Cost Edge 이고 추가했을 때 Cycle을 형성하지 않으므로 Add 합니다.
 5. 이제 (0,1)이 Lowest Cost Edge 이고 추가했을 때 Cycle을 형성하게 되므로 추가하지 않습니다.
 6. 이제 (1,2)가 Lowest Cost Edge 이고 추가했을 때 Cycle을 형성하게 되므로 추가하지 않습니다.
 7. 이제 (2,3)이 Lowest Cost Edge 이고 추가했을 때 Cycle을 형성하지 않으므로 Add 합니다.
- 그래프의 Vertex 개수는 6개인데 현재까지 추가된 Edge가 5개이므로 7단계에서 마치게 됩니다.