
문제 2. 크루스칼(Kruskal) 알고리즘을 이용하여 주어진 그래프의 최소 비용 신장 트리를 구하고자 한다. [배점 : 25점]

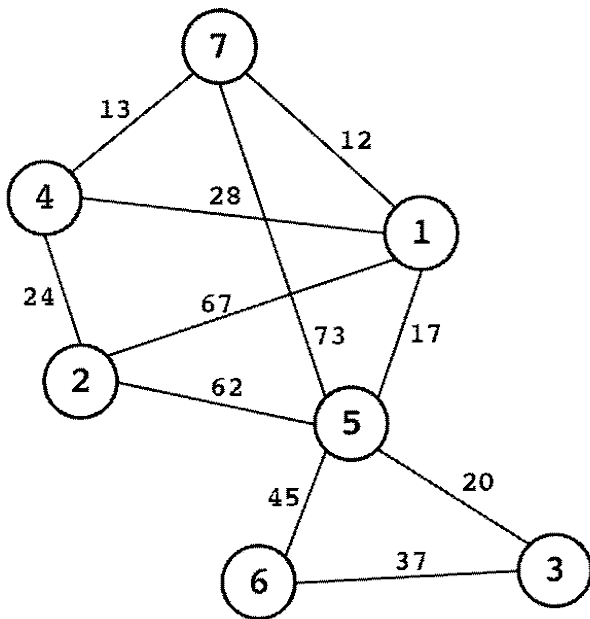
[크루스칼 알고리즘 설명]

크루스칼(Kruskal) 알고리즘은 가장 적은 비용으로 그래프의 모든 노드를 연결하기 위한 최소 비용 신장 트리를 만드는 대표적인 알고리즘이다. 즉, 여러 개의 도시가 있을 때 각 도시의 도로를 이용해 모두 연결하고자 할 때 최소한의 비용으로 하고자 할 때 사용한다. 아래의 설명은 이러한 최소 비용과 함께 사이클이 없는 신장 트리를 구성하는 진행과정을 보여주고 있다.

노드=정점=도시 : 그래프에서 동그라미 부분

간선=거리=비용 : 그래프에서 선 부분

주의) 크루스칼 알고리즘의 간선 숫자 = 노드 숫자 -1



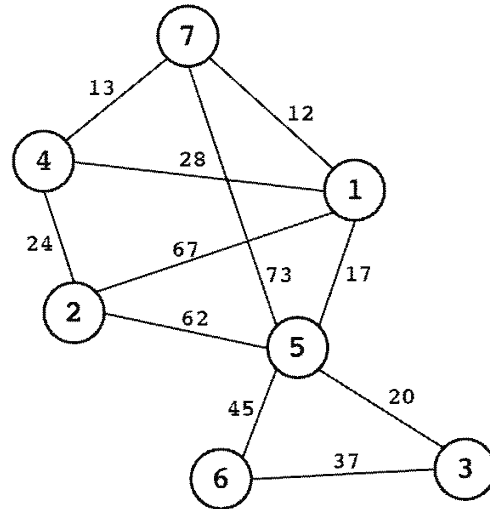
노드1	<table><tr><td>1</td><td>7</td></tr><tr><td>12</td><td></td></tr></table>	1	7	12		<table><tr><td>1</td><td>4</td></tr><tr><td>28</td><td></td></tr></table>	1	4	28		<table><tr><td>1</td><td>2</td></tr><tr><td>67</td><td></td></tr></table>	1	2	67		<table><tr><td>1</td><td>5</td></tr><tr><td>17</td><td></td></tr></table>	1	5	17						
1	7																								
12																									
1	4																								
28																									
1	2																								
67																									
1	5																								
17																									
노드2	<table><tr><td>2</td><td>4</td></tr><tr><td>24</td><td></td></tr></table>	2	4	24		<table><tr><td>2</td><td>5</td></tr><tr><td>62</td><td></td></tr></table>	2	5	62		<table><tr><td>2</td><td>1</td></tr><tr><td>67</td><td></td></tr></table>	2	1	67											
2	4																								
24																									
2	5																								
62																									
2	1																								
67																									
노드3	<table><tr><td>3</td><td>5</td></tr><tr><td>20</td><td></td></tr></table>	3	5	20		<table><tr><td>3</td><td>6</td></tr><tr><td>37</td><td></td></tr></table>	3	6	37																
3	5																								
20																									
3	6																								
37																									
노드4	<table><tr><td>4</td><td>7</td></tr><tr><td>13</td><td></td></tr></table>	4	7	13		<table><tr><td>4</td><td>1</td></tr><tr><td>28</td><td></td></tr></table>	4	1	28		<table><tr><td>4</td><td>2</td></tr><tr><td>24</td><td></td></tr></table>	4	2	24											
4	7																								
13																									
4	1																								
28																									
4	2																								
24																									
노드5	<table><tr><td>5</td><td>6</td></tr><tr><td>45</td><td></td></tr></table>	5	6	45		<table><tr><td>5</td><td>7</td></tr><tr><td>73</td><td></td></tr></table>	5	7	73		<table><tr><td>5</td><td>1</td></tr><tr><td>17</td><td></td></tr></table>	5	1	17		<table><tr><td>5</td><td>3</td></tr><tr><td>20</td><td></td></tr></table>	5	3	20		<table><tr><td>5</td><td>6</td></tr><tr><td>45</td><td></td></tr></table>	5	6	45	
5	6																								
45																									
5	7																								
73																									
5	1																								
17																									
5	3																								
20																									
5	6																								
45																									
노드6	<table><tr><td>6</td><td>3</td></tr><tr><td>37</td><td></td></tr></table>	6	3	37		<table><tr><td>6</td><td>5</td></tr><tr><td>45</td><td></td></tr></table>	6	5	45																
6	3																								
37																									
6	5																								
45																									
노드7	<table><tr><td>7</td><td>1</td></tr><tr><td>12</td><td></td></tr></table>	7	1	12		<table><tr><td>7</td><td>4</td></tr><tr><td>13</td><td></td></tr></table>	7	4	13		<table><tr><td>7</td><td>5</td></tr><tr><td>73</td><td></td></tr></table>	7	5	73											
7	1																								
12																									
7	4																								
13																									
7	5																								
73																									

주어진 그래프 : 총 노드는 7개, 총 간선은 11개

정리 1단계 : 노드 간선 정보 초기화를 위해 간선 정보를 정리함(중복 간선은 회색으로 표시하고 남은 간선정보는 11개임)

1	7	4	7	1	5	3	5	2	4	1	4	3	6	5	6	2	5	1	2	5	7
12	13	17	20	24	28	37	45	62	67	73											

그래프 상태



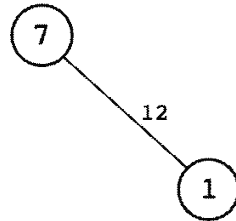
사이클 테이블

1	2	3	4	5	6	7
1	2	3	4	5	6	7

정리 2단계 : 간선 정보를 비용이 낮은 간선부터 오름차순으로 정렬함(사이클 형성 여부를 확인하도록 합집합찾기(Union_Find) 알고리즘을 사용하며, 이를 위해 사이클 테이블 노드 값을 설정함)

1	7	4	7	1	5	3	5	2	4	1	4	3	6	5	6	2	5	1	2	5	7
12	13	17	20	24	28	37	45	62	67	73											

그래프 상태



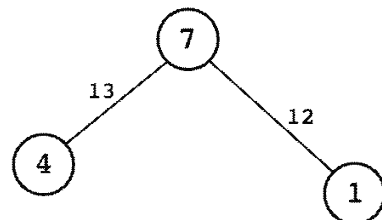
사이클 테이블

1	2	3	4	5	6	7
1	2	3	4	5	6	1

크루스칼 1단계 : 최소 비용을 위해 간선 비용이 가장 낮은 정렬된 첫 번째 간선인 1-7을 선택하고 노드 7이 노드1과 연결되므로 부모 노드로 1을 설정함

1	7	4	7	1	5	3	5	2	4	1	4	3	6	5	6	2	5	1	2	5	7
12	13	17	20	24	28	37	45	62	67	73											

그래프 상태



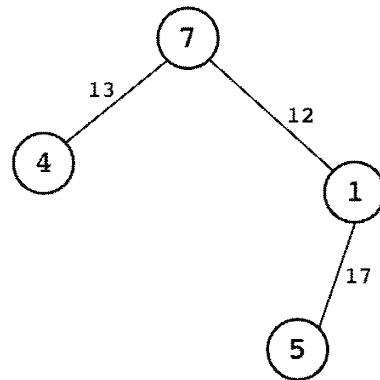
사이클 테이블

1	2	3	4	5	6	7
1	2	3	1	5	6	1

크루스칼 2단계 : 두 번째 간선 4-7을 선택하면 노드4의 부모 노드는 7이지만 다시 7의 부모가 1이므로 역시 1로 변경됨(최상위 부모 노드를 찾아가도록 재귀적으로 구현함)

1	7	4	7	1	5	3	5	2	4	1	4	3	6	5	6	2	5	1	2	5	7
12	13	17	20	24	28	37	45	62	67	73											

그래프 상태



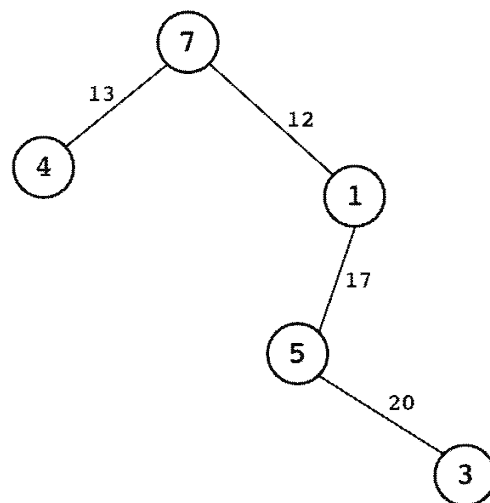
사이클 테이블

1	2	3	4	5	6	7
1	2	3	1	1	6	1

크루스칼 3단계 : 세 번째 간선 1-5를 선택하면 노드 5의 부모 노드가 1로 변경됨

1	7	4	7	1	5	3	5	2	4	1	4	3	6	5	6	2	5	1	2	5	7
12	13	17	20	24	28	37	45	62	67	73											

그래프 상태



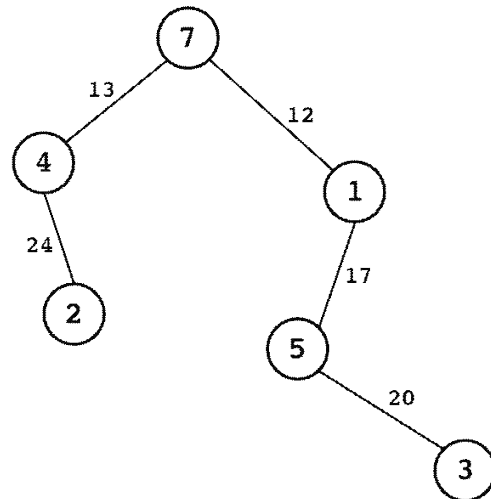
사이클 테이블

1	2	3	4	5	6	7
1	2	1	1	1	6	1

크루스칼 4단계 : 네 번째 간선 3-5를 선택하면 노드3의 부모 노드는 5이지만 5의 부모 노드가 1이므로 1로 설정함(최상위 부모 노드를 찾아가도록 재귀적으로 구현함)

1	7	4	7	1	5	3	5	2	4	1	4	3	6	5	6	2	5	1	2	5	7
12	13	17	20	24	28	37	45	62	67	73											

그래프 상태



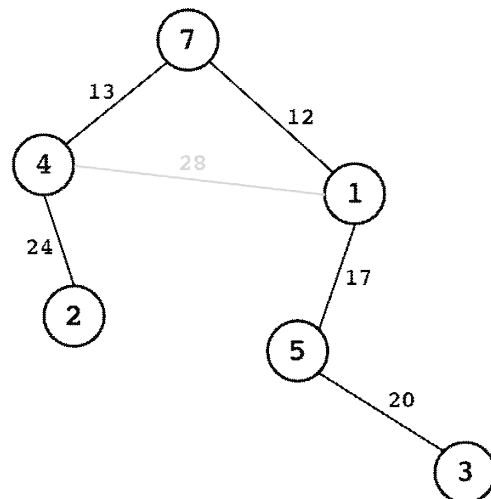
사이클 테이블

1	2	3	4	5	6	7
1	1	1	1	1	6	1

크루스칼 5단계 : 이전 단계처럼 상위 부모 노드를 찾고 노드2의 부모 노드로 1을 찾고 설정함

1	7	4	7	1	5	3	5	2	4	1	4	3	6	5	6	2	5	1	2	5	7
12	13	17	20	24	28	37	45	62	67	73											

그래프 상태



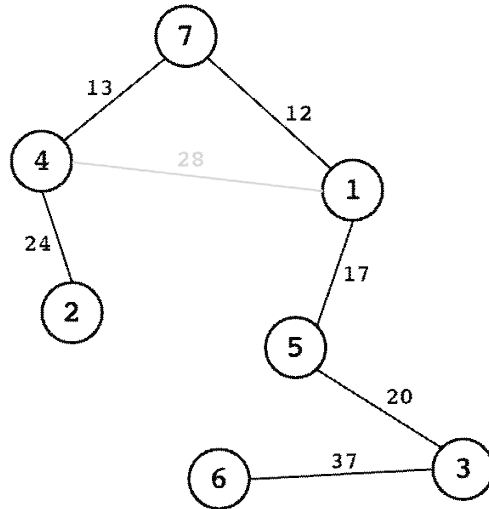
사이클 테이블

1	2	3	4	5	6	7
1	1	1	1	1	6	1

크루스칼 6단계 : 다음 간선인 1-4의 부모는 이미 1로 찾아진 상태이므로 사이클을 형성함을 알 수 있어 신장트리에서 제외함(부모 노드를 찾아서 같은지 아닌지 판정하도록 구현함)

1	7	4	7	1	5	3	5	2	4	1	4	3	6	5	6	2	5	1	2	5	7
12	13	17	20	24	28	37	45	62	67	73											

그래프 상태



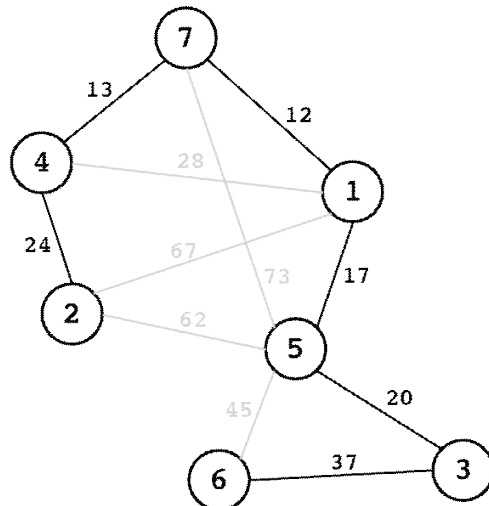
사이클 테이블

1	2	3	4	5	6	7
1	1	1	1	1	1	1

크루스칼 7단계 : 다음 간선인 3-6의 부모 노드를 1로 설정함(크루스칼 알고리즘에서 찾는 최소 비용 신장 트리의 간선은 노드수 - 1 = 노드7개 - 1 = 간선 6개를 다 찾음)

1	7	4	7	1	5	3	5	2	4	1	4	3	6	5	6	2	5	1	2	5	7
12	13	17	20	24	28	37	45	62	67	73											

그래프 상태



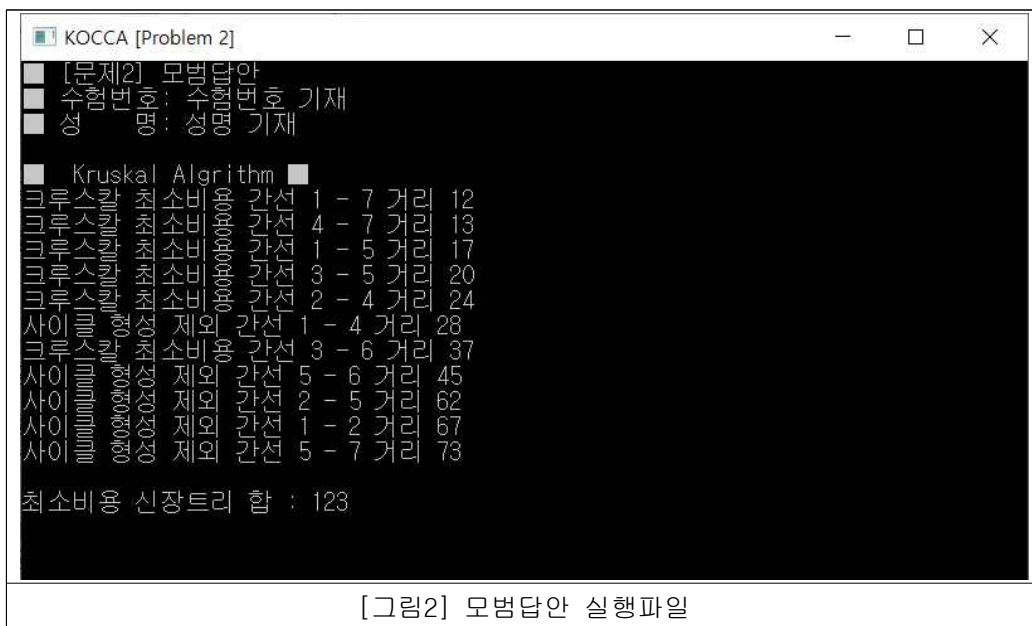
사이클 테이블

1	2	3	4	5	6	7
1	1	1	1	1	1	1

크루스칼 8단계 : 나머지 간선의 부모 노드가 이미 찾아진 상태이므로 제외함

[문제 설명]

- ◎ 주어진 프로그램은 크루스칼 알고리즘을 구현하고 있다.
- ◎ 본 문제는 소스코드에서 표시된 “/* ★ */” 영역을 수정하여 크루스칼 알고리즘이 올바르게 작동할 수 있도록 채워 넣어야 한다.(주석문을 지우고 작성할 것)
 - 주어진 모범 답안을 실행해 보면 [그림2]와 같은 결과화면으로 출력되며, 출력 결과가 일치될 수 있도록 [그림1]을 수정하여 프로그램을 작성해야 한다.
 - 수정해야할 소스는 2-1), 2-2), 2-3)으로 총 3부분을 수정해야 한다.
 - 참고 : 위의 크루스칼 진행과정의 설명 내용과 제공되는 소스의 주석을 적극적으로 참조하여 구현한다.



[제공 자료]

◎ 실행파일(참고용)

- Kruskal_문제풀이.exe : 참고용 실행파일([그림1] 문제풀이 실행파일 >> 수정할 문제)
- Kruskal_모범답안.exe : 참고용 실행파일([그림2] 모범답안 실행파일 결과화면 나옴)

◎ 문제(답안 작성용) 폴더

- 응시자가 작성해야 하는 프로그램
- 프로젝트-소스의 ‘★’ 부분을 확인, 참고할 것.
- 작성 환경 및 라이브러리 : VisualStudio 2017