

백준 1753, 최단거리

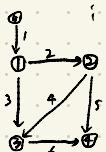
기워드
다익스트라 알고리즘
크스피팅

정의

5(경점 A) 6(경점 B)

1(시작점)

5(경점 A) 1(경점 B) 1(경점 C)



Step 1 최단거리를 나타내는 dist 배열을 무한대로 초기화

INF	INF	INF	INF	INF
1	2	3	4	5

Step 2 dist[시작점, 1] 을 0으로 초기화

0	INF	INF	INF	INF
1	2	3	4	5

Step 3 dist[] 가 1→2 거리를 따른 값을 빼고 dist[2] 와 비교하여 작은 값으로 업데이트해 2를 최단거리를 빼면 3과 4를 업데이트 한다.

예제, 2를 최단거리를 뺀 때 5와 4인 경우에 대해서 3과 4에 같은 1→2를 push 한다.

0	2	INF	INF	INF
1	2	3	4	5

Step 4. 경점 A가 D이 될 때까지 Step 3를 반복

0	2	3	1	INF
1	2	3	4	5

백준 11675, 태일마인

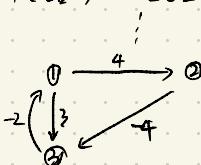
기워드

백만·포도 알고리즘

정의

3(경점 A) 4(경점 B)

1(경점 A) 2(경점 B) 4(경점 C)



Step 1 dist 배열 (1부터 4 경점 까지 걸리는 시간) 을 무한대로 초기화

INF	INF	INF
1	2	3

Step 2 dist[시작점, 1] 을 0으로 초기화

0	INF	INF
1	2	3

Step 3 경선속도 × 경점 수 만큼 가중치를 더한 값을 적은 값으로 계속 업데이트
한 dist가 INF는 경우 업데이트 개선되어 처리

0	INF	INF	→	0	9	INF	→	0	4	3
1	2	3		1	2	3		1	2	3

Step 4 마지막으로 경점 4 번째를 업데이트 하는데 기본값보다 작아지면 사이클이 존재

0	4	3	→	0	0	3
						3

같이 출력적으로 흐름 차이를 존재

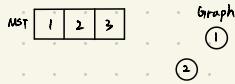
백준 1197, 최소 스패닝 트리

Step 1

Input : 원소 개수와 원소
VEC : 원소 간의 연결을 표현하는 Vector

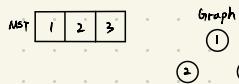
SUM : 원소를 더하는 함수

vec	<table border="1"> <tr><td>1</td><td>2</td><td>1</td><td>3</td><td>2</td><td>3</td></tr> <tr><td>1</td><td></td><td>3</td><td></td><td>2</td><td></td></tr> </table>	1	2	1	3	2	3	1		3		2		(정렬된 원소로 초기화)
1	2	1	3	2	3									
1		3		2										



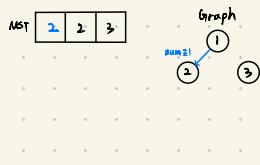
Step 2 모든 원소가 MST에 포함되는지 확인

vec	<table border="1"> <tr><td>1</td><td>2</td><td>2</td><td>3</td><td>1</td><td>3</td></tr> <tr><td>1</td><td></td><td>2</td><td></td><td>3</td><td></td></tr> </table>	1	2	2	3	1	3	1		2		3	
1	2	2	3	1	3								
1		2		3									



Step 3 MST에 포함되지 않은 원소들 중에서 MST에 더하기 위해 선택되는 원소는?

vec	<table border="1"> <tr><td>1</td><td>2</td><td>2</td><td>3</td><td>1</td><td>3</td></tr> <tr><td>1</td><td></td><td>2</td><td></td><td>3</td><td></td></tr> </table>	1	2	2	3	1	3	1		2		3	
1	2	2	3	1	3								
1		2		3									



Union-Find 구현

```
int find(int a) // a가 루트인가?
{
    if (MST[a] == a)
        return MST[a];
    else
        return MST[a] = find(MST[a]);
```

```
void Union(int a, int b)
```

```
{
    a = find(a);
    b = find(b);
    if (a != b) // a와 b가 다른 집합인 경우
        MST[a] = b; // a와 b를 합침
}
```

Step 4 MST에 포함되는 원소를 더한 결과는?

vec	<table border="1"> <tr><td>1</td><td>2</td><td>2</td><td>3</td><td>1</td><td>3</td></tr> <tr><td>1</td><td></td><td>2</td><td></td><td>3</td><td></td></tr> </table>	1	2	2	3	1	3	1		2		3	
1	2	2	3	1	3								
1		2		3									



Step 4 모든 원소가 MST에 포함되는지 확인하는 계산 결과는?

vec	<table border="1"> <tr><td>1</td><td>2</td><td>2</td><td>3</td><td>1</td><td>3</td></tr> <tr><td>1</td><td></td><td>2</td><td></td><td>3</td><td></td></tr> </table>	1	2	2	3	1	3	1		2		3	
1	2	2	3	1	3								
1		2		3									

