

Graph

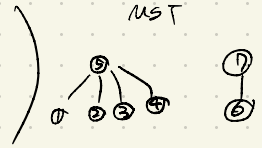
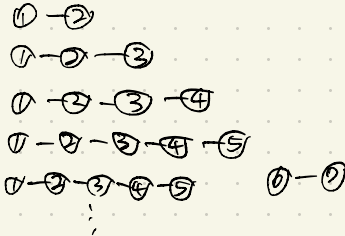


백준 1197, 최소 신장 트리

입력

1 2	step 1	1 2
2 3	step 2	2 3
3 4	step 3	3 4
1 5	step 4	1 5
6 7	step 5	6 7
...		

Union-Find



pseudocode

```

int Find (int n)
{
    if (arr[n] == n)
        return arr[n];
    return Find (arr[n]);
}

```

void Union (int a, int b)

```

{
    a = Find (a);
    b = Find (b);
    if (a != b)
    {
        arr[b] = a;
    }
}

```

백준 1753, 최단 경로

자료 구조
인접 리스트
Priority-queue (힙)

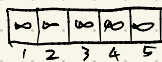
자료 구조
인접 리스트
priority-queue (힙)

결과

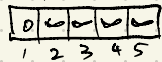
→ (정점 A) 6번선 A)
1 (A가 정점)
→ (정점 A) 1 (정점 B) 1 (A > B)
정점 A) 1 (정점 B) 1 (A > B)
정점 A) 1 (정점 B) 1 (A > B)

풀이

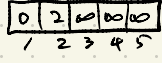
Step 1 시작점에서 각 정점의 가중치 초기화



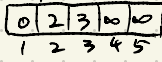
Step 2 시작점 1을 우선 순위화



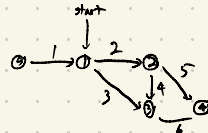
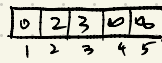
Step 3 1에서 붙어있는 정점 2, 3, 4, 5를 우선 순위화 (가중치 초기화)



Step 4 1에서 붙어있는 정점 3, 4, 5를 우선 순위화



Step 5 현재 2가 정점 2, 2에서 3으로 가는 정점 4를 우선 순위화
현재 3가 정점 3과 비교 후 작은 값을 대입한다



Step 6 정점 4에서 Step 5를 우선 순위화
정점 4에서 Step 5를 우선 순위화
정점 4에서 Step 5를 우선 순위화

