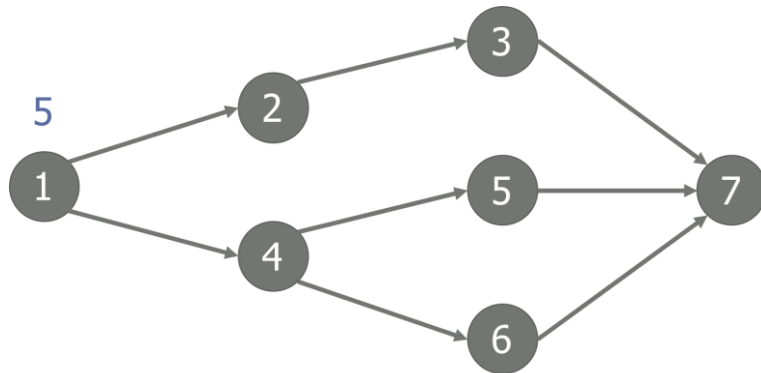


작업

<https://www.acmicpc.net/problem/2056>

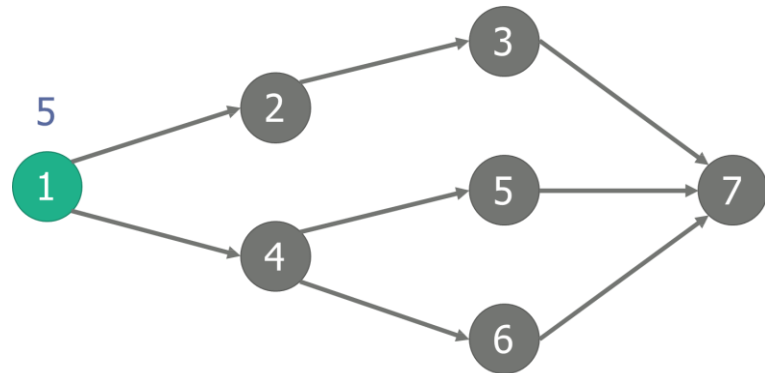
- 작업의 선행 관계가 주어졌을 때 모두 마치는 가장 빠른 시간을 구하면 된다.

큐 : 1 / 현재 작업 : 1



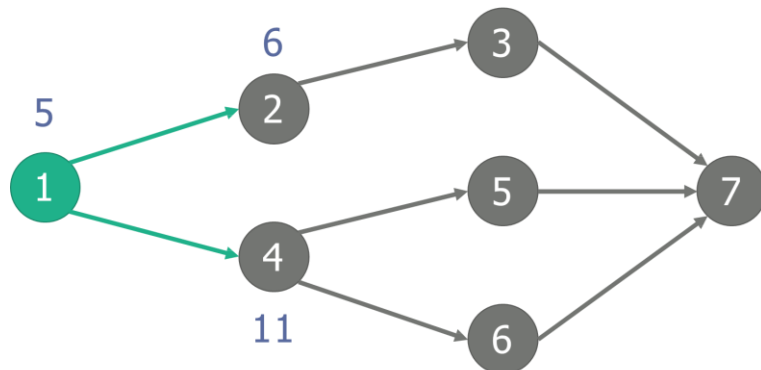
i	1	2	3	4	5	6	7
시간	5	1	3	6	1	8	4

큐 : / 현재 작업 : 1

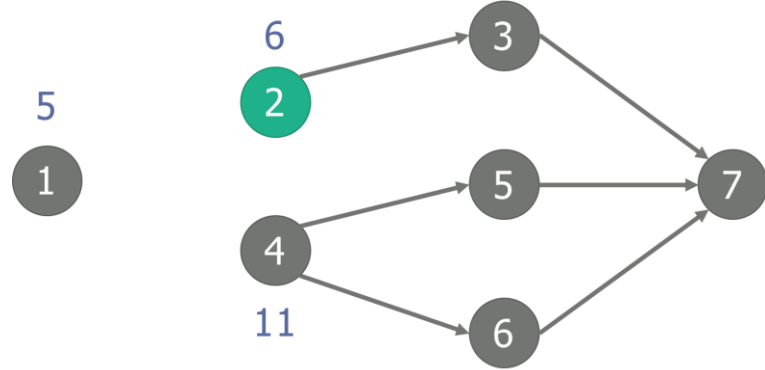


i	1	2	3	4	5	6	7
시간	5	1	3	6	1	8	4

큐 : 2, 4 / 현재 작업 : 1



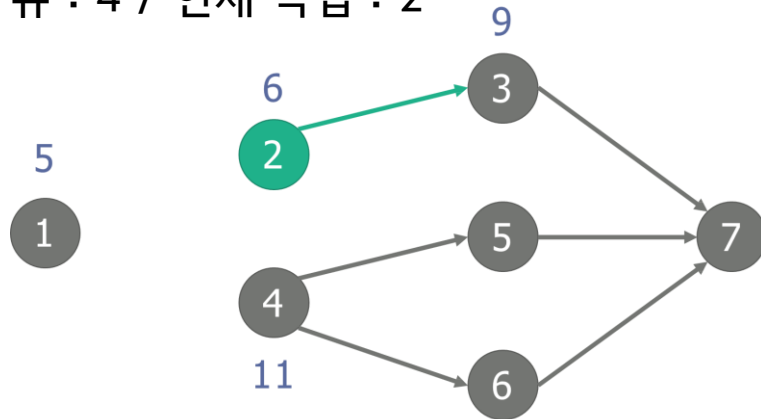
큐 : 4 / 현재 작업 : 2



작업

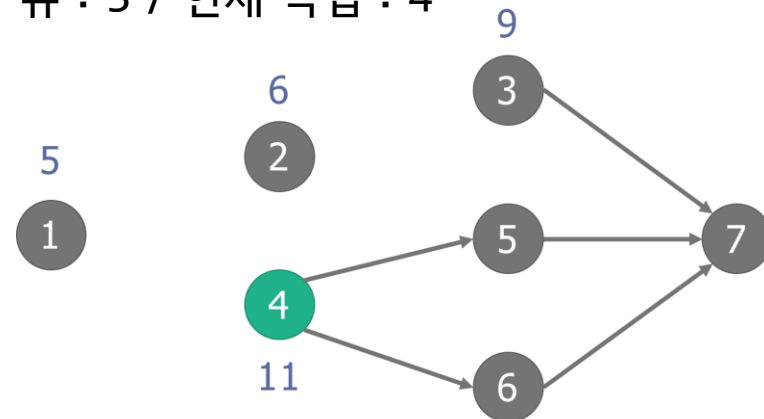
<https://www.acmicpc.net/problem/2056>

큐 : 4 / 현재 작업 : 2



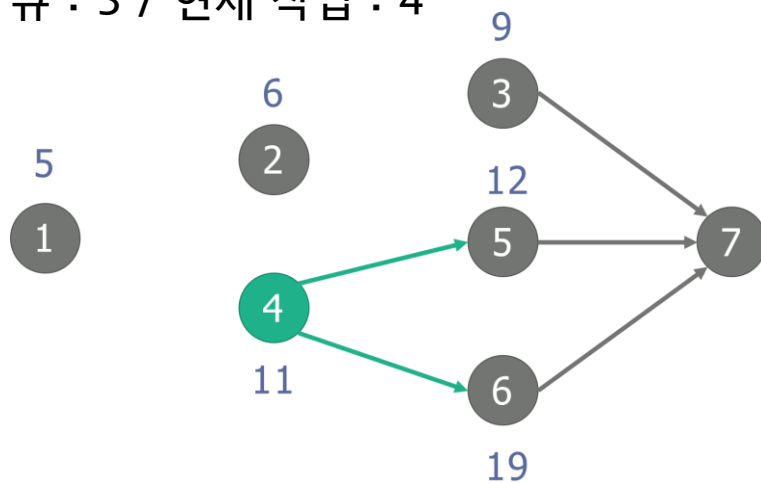
i	1	2	3	4	5	6	7
시간	5	1	3	6	1	8	4

큐 : 3 / 현재 작업 : 4

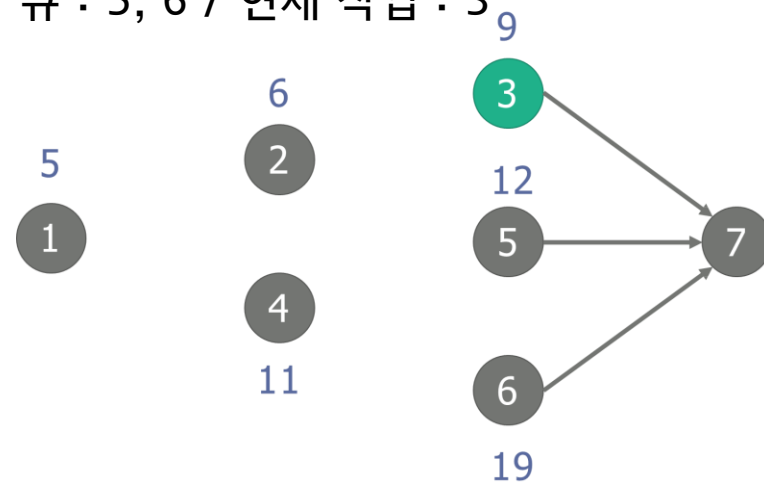


i	1	2	3	4	5	6	7
시간	5	1	3	6	1	8	4

큐 : 3 / 현재 작업 : 4



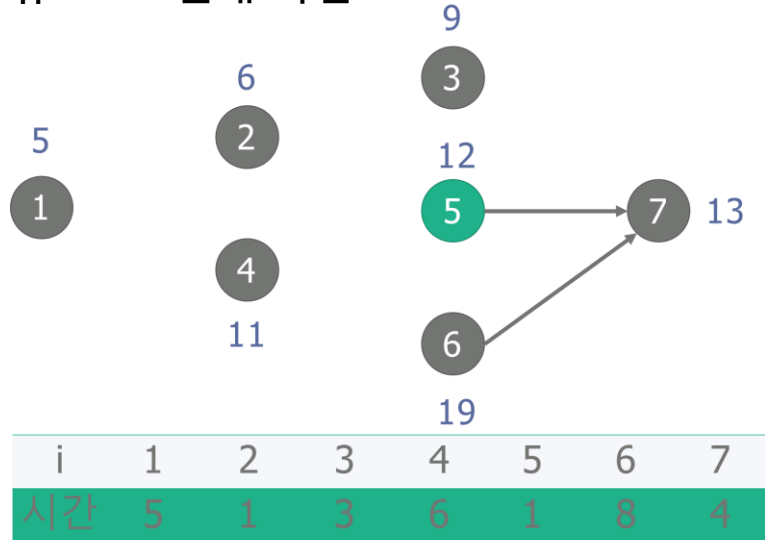
큐 : 5, 6 / 현재 작업 : 3



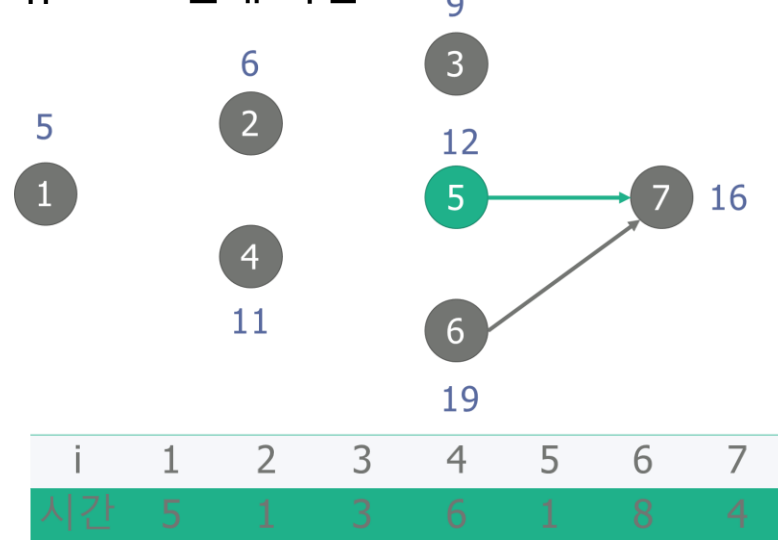
작업

<https://www.acmicpc.net/problem/2056>

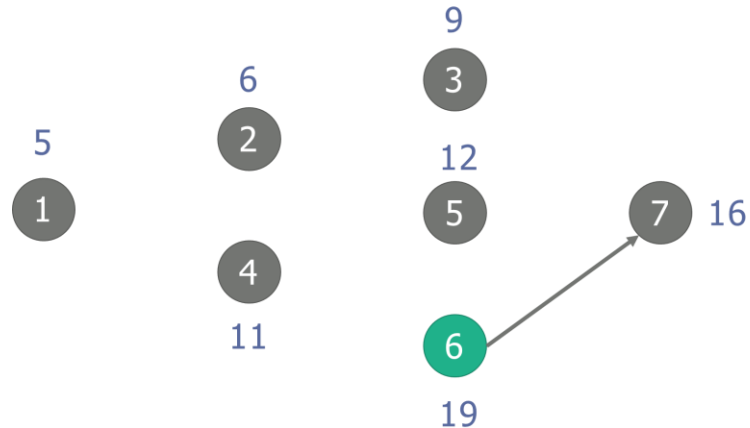
큐 : 6 / 현재 작업 : 5



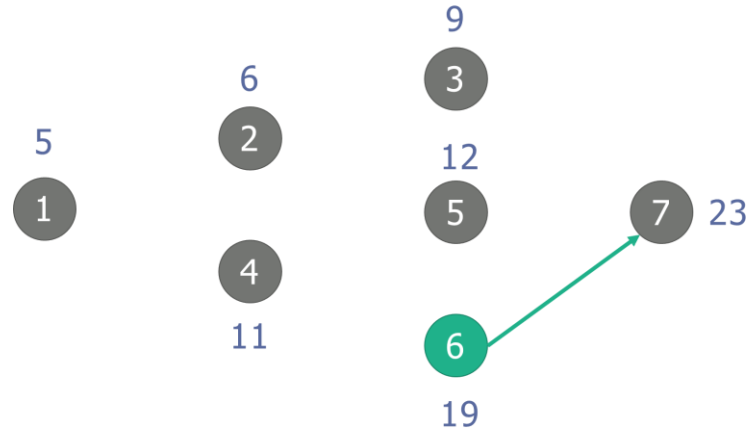
큐 : 6 / 현재 작업 : 5



큐 : / 현재 작업 : 6



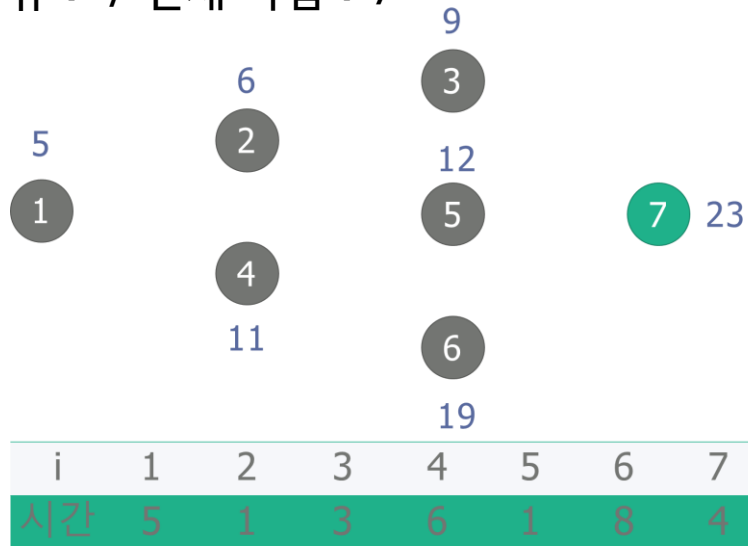
큐 : / 현재 작업 : 6



작업

<https://www.acmicpc.net/problem/2056>

큐 : / 현재 작업 : 7



큐 : / 현재 작업 :



작업

<https://www.acmicpc.net/problem/2056>

```
queue<int> q;
for (int i = 1; i <= N; i++) {
    if (ind[i] == 0) {
        q.push(i);
        dist[i] = work[i];
    }
}

while(!q.empty()){
    int u = q.front();
    q.pop();
    for (int i = 0; i<AdjList[u].size(); i++) {
        int v = AdjList[u][i];
        ind[v] -= 1;
        if (dist[v] < dist[u] + work[v]) {
            dist[v] = dist[u] + work[v];
        }
        if (ind[v] == 0) {
            q.push(v);
        }
    }
}
```

```
int ans = 0;
for (int i = 1; i <= N; i++) {
    if (ans < dist[i]) {
        ans = dist[i];
    }
}
```