Week 14

2024-10-03

장난감 조립

장난감 완제품을 만드는 데
몇 번째 기본 부품이 얼마나 필요한지 ??

완제품의 번호는 n 이고 m 개의 간선 정보가 주어진다 .

간선 정보는 t1, t2, t3 식으로 주어지고
t1 을 만들기 위해 t2 가 t3 개 필요하다는 의미를 가진다 .

장난감 조립

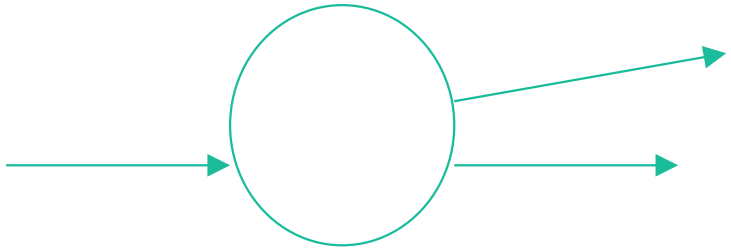
특정 부품을 만들기 위해서는 다른 어떤 부품이 필요하다

부품을 그래프의 노드로 생각하고 , 필요 관계를 간선으로 생각하면

그래프 노드간의 순서가 주어졌기 때문에 위상 정렬을 사용해 볼 수 있다 .

장난감 조립

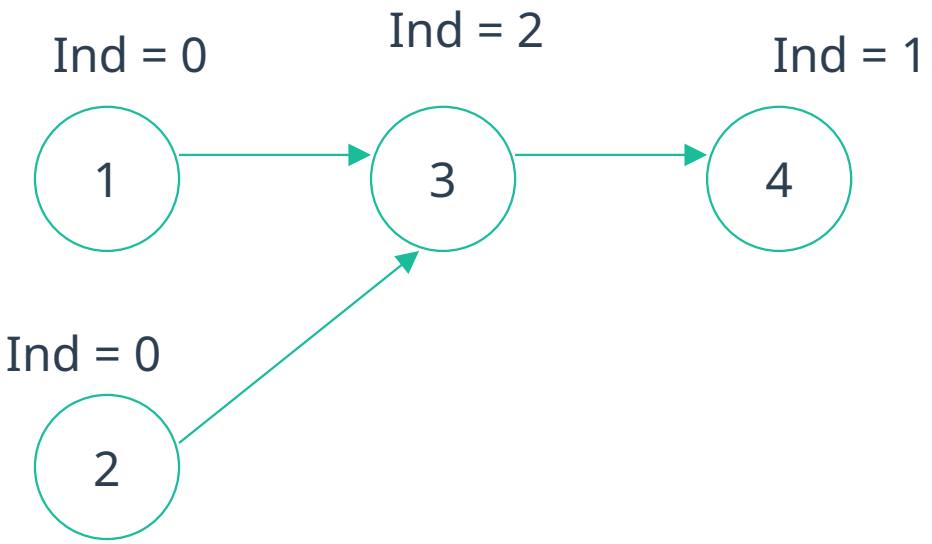
노드로 들어오는 간선의 개수를 in degree 라고 하고
노드에서 바깥으로 나가는 간선의 개수를 out degree 라고 한다



In degree = 1

Out degree = 2

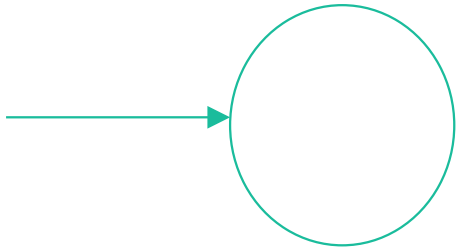
장난감 조립



4
3
1 3
2 3
3 4

장난감 조립

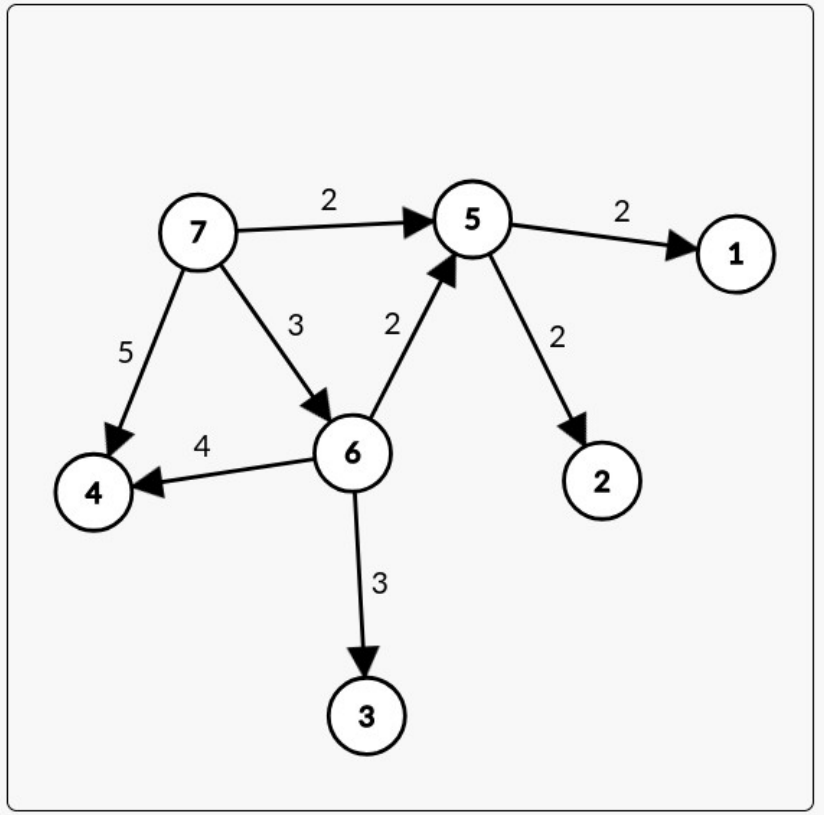
Out degree 가 0 인 부품이 기본 부품이 된다 .
그 부품을 만들기 위해 다른 부품이 필요하지 않다는 뜻이므로



In degree = 1

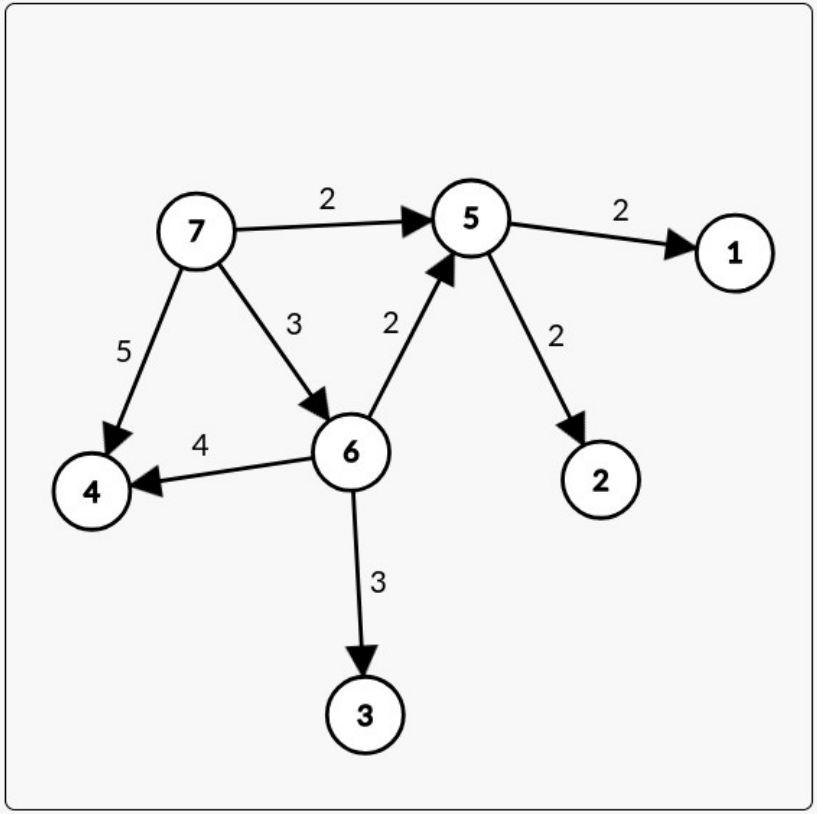
장난감 조립

```cpp
for(int i = 0; i < m; i++) {
    int t1, t2, t3;
    cin >> t1 >> t2 >> t3;
    g[t1].emplace_back(t2, t3);
    ind[t2] += 1;
    outd[t1] += 1;
}
```



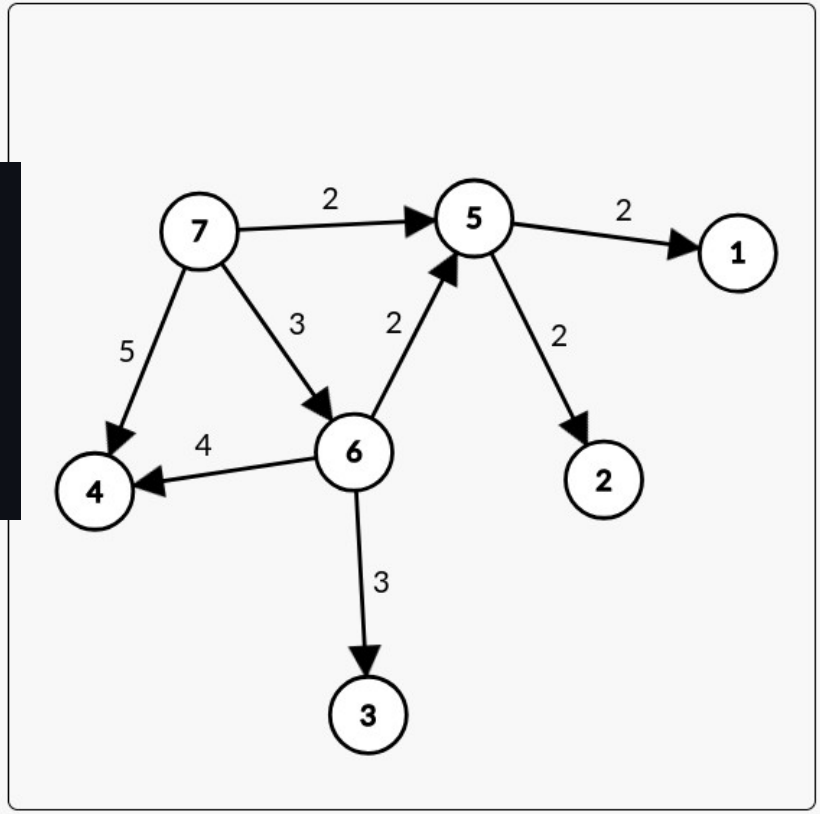
장난감 조립

```
queue<int> q;
q.push(n);
cnt[n] = 1;
while(!q.empty()) {
    int now = q.front();
    q.pop();
    for(auto [next, cost] : g[now]) {
        ind[next]--;
        cnt[next] += cnt[now] * cost;
        if(ind[next] == 0) {
            q.push(next);
        }
    }
}
```



장난감 조립

```
for(int i = 1; i < n; i++) {
    if(outd[i] == 0 && cnt[i] != 0)  {
        cout << i << ' ' << cnt[i] << '\n';
    }
}
```



중앙값 구하기

1. 입력이 홀수이기 때문에 중앙값의 개수는 무조건 (n + 1) / 2

MAX HEAP 과 MIN HEAP 을 만들어서 ,
왼쪽 부분과 오른쪽 부분을 담는다 .
왼쪽 힙의 맨 위에 중앙값이 있다고 가정

1. l 사이즈가 r + 1 보다 크다면 왼쪽껄 오른쪽으로 이동
2. l.top() 이 r.top() 보다 크다면 하나씩 교환

중앙값 구하기

4
3
2
1

6
7
8
9

```cpp
void solve() {
    int n; cin >> n;
    cout << (n + 1) / 2 << '\n';
    priority_queue<int> l;
    priority_queue<int, vector<int>, greater<int>> r;
    for(int i = 0; i < n; i++) {
        int x; cin >> x;
        l.push(x);
        while(l.size() > r.size() + 1) {
            r.push(l.top());
            l.pop();
        }
        while(!l.empty() && !r.empty() && l.top() > r.top()) {
            r.push(l.top()); l.pop();
            l.push(r.top()); r.pop();
        }
        if(i % 2 == 0) {
            cout << l.top() << ' ';
        }
    }
    cout << '\n';
    return;
}
```