

Tree DP

Lectured by 김경한

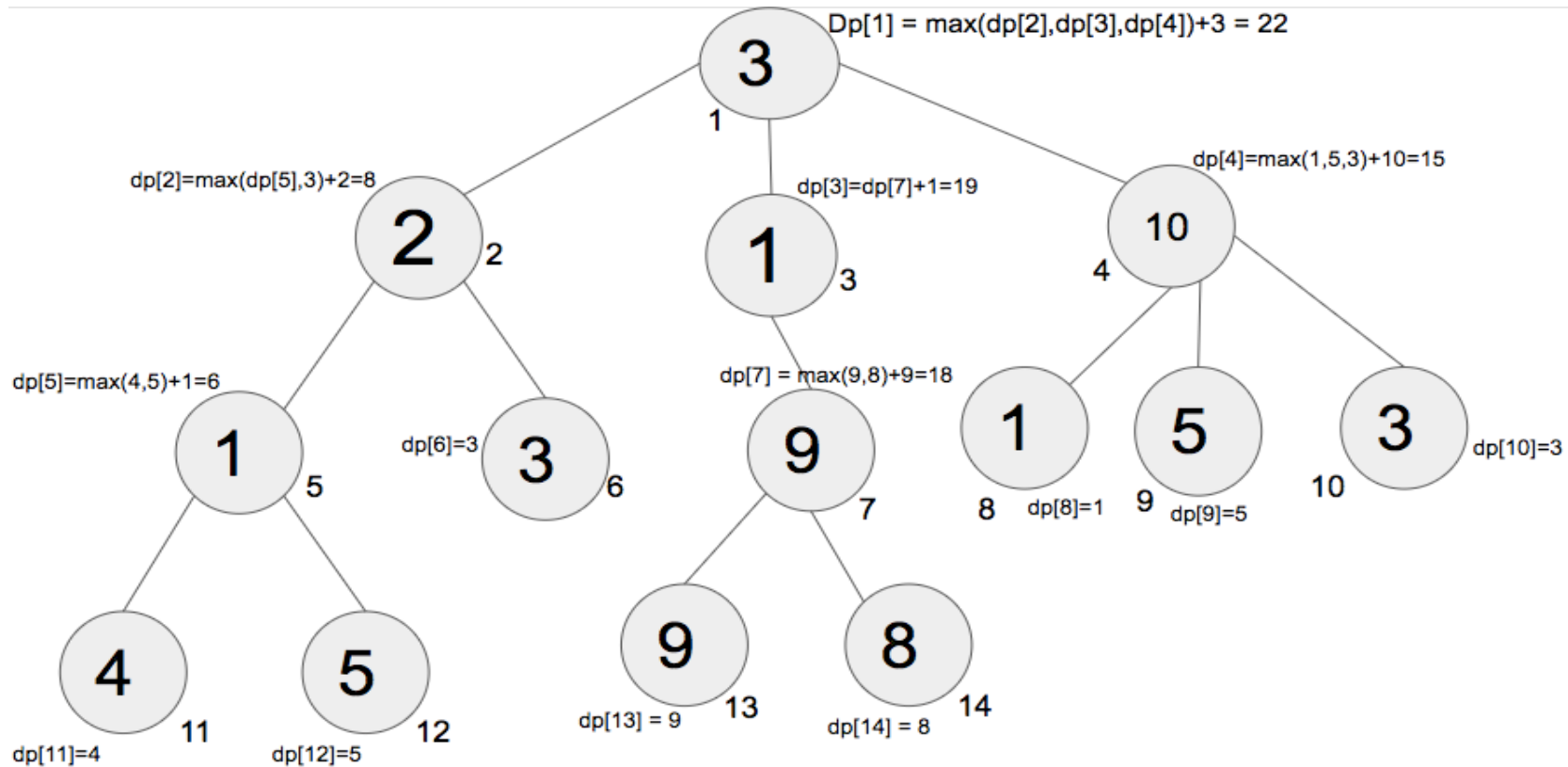
일반적인 DP

- 예시 : 피보나치 수열
- 점화식 : $DP[i] = DP[i - 1] + DP[i - 2]$, $DP[0] = 0$, $DP[1] = 1$

i	0	1	2	3	4	5	6	7
DP[i]	0	1	1	2	3	5	8	13

Tree DP

- 정보를 저장하는 자료구조가 배열(Array)에서 트리(Tree)로 바뀐 형태



예시


- 트리의 모든 노드에 비용이 주어졌을 때,
루트 노드에서부터 리프 노드까지 이어지는 경로 중 최댓값
- u 가 부모 노드, v_1, v_2, \dots, v_n 가 자식 노드라고 할 때
$$DP[u] = \text{MAX}(DP[v_1], DP[v_2], \dots, DP[v_n]) + \text{cost}(u)$$
- 주로 DFS로 경로를 탐색하며 DP 배열을 채웁니다.

예시 코드


```
1 // C++ code to find the maximum path sum
2 #include <bits/stdc++.h>
3 using namespace std;
4 using vi = vector<int>;
5 using vvi = vector<vi>;
6
7 vi cost, memo;
8 vvi adj;
9
10 void dfs(int node, int parent) {
11     memo[node] = cost[node];
12
13     int maximum = 0;
14     for (int child : adj[node]) {
15         if (child == parent) continue;
16         dfs(child, node);
17         maximum = max(maximum, memo[child]);
18     }
19
20     memo[node] += maximum;
21 }
22
23 int maximum_value(int root) {
24     dfs(root, 0);
25     return memo[root];
26 }
27
```


```
28 int main() {
29     int n, root;
30     cin >> n >> root;
31
32     cost.resize(n + 1);
33     memo.resize(n + 1);
34     adj.resize(n + 1);
35
36     for (int i = 1; i <= n; i++)
37         cin >> cost[i];
38
39     int s, e;
40     for (int i = 1; i < n; i++) {
41         cin >> s >> e;
42         adj[s].push_back(e);
43         adj[e].push_back(s);
44     }
45
46     cout << maximum_value(root) << '\n';
47
48     return 0;
49 }
50
```

연습 문제


 15861번. 트리와 쿼리


 2533번. 사회망 서비스(SNS)


 1623번. 신년 파티


 1949번. 우수 마을


 2213번. 트리의 독립집합

 1135번. 뉴스 전하기

 17831번. 대기업 승범이네

 1693번. 트리 색칠하기

 15647번. 로스팅하는 엠마도
바리스타입니다.

 1289번. 트리의 가중치