“호숫가의 개미굴” 문제 풀이

작성자: 나정휘

서술의 편의를 위해 개미굴의 방을 정점, 두 방을 연결하는 통로를 간선으로 표기한다. 즉, 문제에서 주어진 개미굴은 N개의 정점으로 구성된 사이클, 그리고 사이클을 구성하는 각 정점에 C_i개의 정점이 추가로 달려있는 그래프로 생각할 수 있다.

부분문제 1

다음과 같은 4가지 경우에 개미굴에 살 수 있는 개미의 최댓값을 모두 구한 뒤, 그 중 최댓값을 출력하면 된다.

1. 1번 정점과 2번 정점에 모두 개미가 살지 않는 경우
2. 1번 정점에 개미가 살지만 2번 정점에 개미가 살지 않는 경우
3. 1번 정점에 개미가 살지 않지만 2번 정점에 개미가 사는 경우
4. 1번 정점과 2번 정점에 모두 개미가 사는 경우

부분문제 2

N개의 정점으로 구성된 단순 사이클만 있는 상황에서의 정답을 구해야 한다.

짝수 번째 정점에 개미를 한 마리씩 배치하면 $\lfloor \frac{N}{2} \rfloor$마리의 개미를 배치할 수 있고, 이보다 더 많이 배치할 수 없음을 알 수 있다. 따라서 $\lfloor \frac{N}{2} \rfloor$을 출력하면 된다.

부분문제 3

그래프 이론에서 각 간선이 연결하는 두 정점 중 최대 한 정점만 선택할 수 있을 때 정점을 최대한 많이 선택하는 문제를 **최대 독립 집합** 문제라고 부른다. 일반적인 그래프에서는 최대 독립 집합 문제를 다항 시간에 해결할 수 없지만, 트리에서는 동적 계획법을 이용해 $O(N)$ 시간에 해결할 수 있다.

구체적으로, 트리에서 아무 정점을 하나 잡아서 루트로 만든 다음 점화식을 다음과 같이 정의하면, 깊이 우선 탐색을 이용해 점화식을 계산해서 최대 독립 집합의 크기를 구할 수 있다.

- $D(v, 0)$: v번 정점과 그의 자손 정점만 남긴 서브트리에서, v번 정점을 선택하지 않았을 때의 최댓값
- $D(v, 1)$: v번 정점과 그의 자손 정점만 남긴 서브트리에서, v번 정점을 선택했을 때의 최댓값

사이클을 구성하는 간선 중 아무거나 하나를 끊으면 그래프가 트리로 바뀌고, 끊어진 간선 $e = (u, v)$가 연결하고 있는 두 정점 u, v 중 최대 하나만 선택할 수 있을 때의 최대 독립 집합을 구하면 된다. 따라서 이 문제는 트리 T와 트리의 두 정점 u, v가 주어졌을 때 u, v 중 최대 하나만 선택할 수 있는 최대 독립 집합 문제라고 생각할 수 있다.

부분문제 3은 사이클을 구성하는 모든 간선에 대해 동적 계획법을 수행하면 문제를 해결할 수 있다. 사이클을 구성하는 간선 $(i, i+1)$을 끊은 트리에서 i번째 정점을 사용하지 않는 최대 독립 집합을 모든 간선에 대해 구하면, 그 중 최댓값이 원본 그래프의 최대 독립 집합의 크기가 된다. 이때 시간 복잡도는 $O((N + \sum C_i)^2)$이다.

부분문제 5

부분문제 3의 풀이를 확장하면 부분문제 5를 해결할 수 있다. 부분문제 3에서는 총 N개의 트리에서 각각 동적 계획법을 수행했지만, 부분문제 5에서는 단 한 번의 동적 계획법을 이용해 문제를 해결해야 한다.

1번 정점과 N번 정점을 잇는 간선을 끊은 트리에서, 깊이 우선 탐색을 이용해 다음과 같이 정의된 점화식을 계산하면 $O(N + \sum C_i)$번의 연산으로 문제를 해결할 수 있다.

- $D(v, 0, 0)$: v번 정점과 그의 자손 정점만 남긴 서브트리에서, v번 정점을 선택하지 않고 N번 정점도 선택하지 않았을 때의 최댓값
- $D(v, 1, 0)$: v번 정점과 그의 자손 정점만 남긴 서브트리에서, v번 정점을 선택하고 N번 정점을 선택하지 않았을 때의 최댓값
- $D(v, 0, 1)$: v번 정점과 그의 자손 정점만 남긴 서브트리에서, v번 정점을 선택하지 않았을 때의 최댓값
- $D(v, 1, 1)$: v번 정점과 그의 자손 정점만 남긴 서브트리에서, v번 정점을 선택했을 때의 최댓값

부분문제 7

3개 이상의 정점으로 구성된 모든 연결 그래프에는 차수가 1인 모든 정점을 포함하는 최대 독립 집합이 존재하고, 이는 귀류법을 이용해 증명할 수 있다.

차수가 1인 모든 정점을 포함하는 최대 독립 집합이 없다고 가정하면, 최대 독립 집합에 포함되지 않은 차수가 1인 정점 v가 존재할 것이다. v와 인접한 유일한 정점 x는 최대 독립 집합에 포함되어 있어야 하며, 이때 x의 차수는 1보다 크다. 그러므로 x 대신 v를 넣더라도 최대 독립 집합을 만들 수 있다.

따라서 $N + \sum C_i \geq 3$이면 항상 모든 쪽방을 포함하는 정답이 존재하고, 부분문제 7은 $N \geq 2, C_i \geq 1$을 만족하므로 $\sum C_i$를 출력하면 문제를 해결할 수 있다.

부분문제 8

부분문제 8은 부분문제 7의 관찰을 확장해서 해결할 수 있다.

먼저 쪽방이 하나도 없는 경우를 먼저 처리하자. 이는 부분문제 2의 상황과 동일하므로 $\lfloor \frac{N}{2} \rfloor$를 출력하면 된다.

이제 쪽방이 최소한 하나 이상 있는, $N + \sum C_i \geq 3$인 경우만 생각해도 된다. 부분문제 7의 관찰에 의해 모든 쪽방을 포함하는 답안이 존재하므로 $C_i \geq 1$인 모든 정점에 달려 있는 쪽방에 개미를 배치할 수 있다. 그러면 $C_i = 0$인 정점들이 남게 되는데, $C_i, C_{i+1}, C_{i+2}, \cdots, C_{j-1}, C_j$가 모두 0이면 $C_i, C_{i+2}, C_{i+4}, \cdots$와 같이 홀수 번째 정점에 개미를 배치하는 것이 최적임을 알 수 있다. 즉, C_i가 0인 구간에서는 홀수 번째 자리에 개미를 배치하면 되므로, $\lfloor$(구간의 길이 + 1) / 2$\rfloor$를 정답에 더하면 된다. 시간 복잡도는 $O(N)$이다.

C, C++, Java에서 64비트 정수 타입이 아닌 32비트 정수 타입을 사용하면 부분문제 6만 맞게 되고, $C_i = 0$인 구간을 확인하는 부분을 비효율적으로 구현하면 부분문제 4만 맞게 된다.