Centroid Decomposition on Tree

구재현 (koosaga)

http://koosaga.myungwoo.kr

Why Divide and Conquer?

- 분할 정복을 왜 하나요?
- 기존에 분할 정복으로 문제를 풀었던 사례들을 떠올려 보실 수 있나요?

Why Divide and Conquer?

- 분할 정복을 왜 하나요?
- 기존에 분할 정복으로 문제를 풀었던 사례들을 떠올려 보실 수 있나요?
- 정렬, 수열의 반전 수 ($i < j, a_i > a_j$) 세기, 가장 가까운 두 점 찾기 등등...

Why Divide and Conquer?

- 분할 정복을 왜 하나요?
- 기존에 분할 정복으로 문제를 풀었던 사례들을 떠올려 보실 수 있나요?
- 정렬, 수열의 반전 수 $(i < j, a_i > a_j)$ 세기, 가장 가까운 두 점 찾기 등등...
- 보통, $O(n^2)$는 쉽게 생각할 수 있으나, 이를 $O(nlgn)$에 줄이는 게 쉽지 않을 때 자주 사용되는 게 분할 정복입니다.
- ~~그냥 도저히 못 풀겠으면 분할 정복입니다.~~

Why Divide and Conquer in Tree?

- ~~트리 문제들은 대부분 도저히 못 풀겠기 때문입니다.~~
- 예를 들어, 구간의 배열 합이 K인 subarray의 개수를 세는 문제를 생각해 봅시다.
- 1차원 배열에서는, 부분합을 만들면 어렵지 않게 해결할 수 있는 문제입니다. 부분합 배열을 S_i 라고 정의하면
- $1 \leq i \leq j \leq n$ such that $S_i - S_{j-1} = K$
- 의 형태가 나와서, 적당한 전처리를 하면 i가 고정되었을 때 $O(lgn)$에 j의 개수를 구할 수 있기 때문입니다.

Why Divide and Conquer in Tree?

- 트리에서 똑같이 해 봅시다. 루트에서 현재 정점까지의 원소의 합을 똑같이 S_i로 정의하면
- $1 \le i \le j \le n$ such that $S_i + S_j - S_{LCA(i,j)} - S_{par(LCA(i,j))} = K$
- (이 때, $par(i) = i$번 노드의 조상, $LCA(i,j) = i, j$번 노드의 최저 공통 조상)
- 이제 i를 고정시키면 되겠죠?

Why Divide and Conquer in Tree?

- 트리에서 똑같이 해 봅시다. 루트에서 현재 정점까지의 원소의 합을 똑같이 S_i로 정의하면
- $1 \leq i \leq j \leq n$ such that $S_i + S_j - S_{LCA(i,j)} - S_{par(LCA(i,j))} = K$
- ($\text{이 때, } par(i) = i$번 노드의 조상, $LCA(i,j) = i, j$번 노드의 최저 공통 조상)
- 이제 i를 고정시키면 되겠죠?

Centroid Decomposition

- 못 풀겠으면 분할 정복입니다! 한번 시도해 봅시다.
- 그런데, 분할 정복에서는 수열의 중심을 쉽게 잡을 수 있었지만, 트리에서는 그게 쉬워 보이지 않네요.
- 결국, $O(lgn)$ 깊이만 재귀를 돌아서 모든 트리의 정점을 처리하는 게 목표인데
- 트리의 지름의 중심을 잡는건 어떨까요?

Centroid Decomposition

- 못 풀겠으면 분할 정복입니다! 한번 시도해 봅시다.
- 그런데, 분할 정복에서는 수열의 중심을 쉽게 잡을 수 있었지만, 트리에서는 그게 쉬워 보이지 않네요.
- 결국, $O(lgn)$ 깊이만 재귀를 돌아서 모든 트리의 정점을 처리하는 게 목표인데
- 트리의 지름의 중심을 잡는건 어떨까요?
- 잘 되나요? 왜 잘 되지 않나요?

Centroid Decomposition

Lemma 1 : 트리의 Centroid

- 임의의 크기 n의 트리에 대해, 해당 정점을 제거 했을 때, 나머지 트리 조각들의 크기가 $n/2$ 이하인 정점이 존재합니다.
- 트리의 어떠한 정점에 대해서, 그 정점을 제거 했을 때 나머지 트리 조각들의 최대 크기를 $maxSubtree(v)$ 라고 정의합시다.
- $maxSubtree(v)$ 를 최소로 하는 정점이 있는데, 이 정점의 $maxSubtree(v) > n/2$ 라면 어떻게 되나요?

Centroid Decomposition

Lemma 1 : 트리의 Centroid

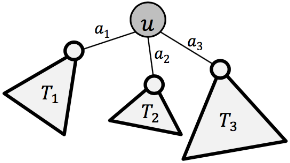
- 임의의 크기 n의 트리에 대해, 해당 정점을 제거 했을 때, 나머지 트리 조각들의 크기가 $n/2$ 이하인 정점이 존재합니다.
- 트리의 어떠한 정점에 대해서, 그 정점을 제거 했을 때 나머지 트리 조각들의 최대 크기를 $maxSubtree(v)$ 라고 정의합시다.
- $maxSubtree(v)$ 를 최소로 하는 정점이 있는데, 이 정점의 $maxSubtree(v) > n/2$ 라면 어떻게 되나요?
- 가정에 모순이 됩니다. 그리고 $maxSubtree(v)$를 최소로 하는 정점은 $maxSubtree(v) \leq n/2$를 만족합니다.
- 이러한 정점을 centroid라고 합니다.

Centroid Decomposition

그러면...

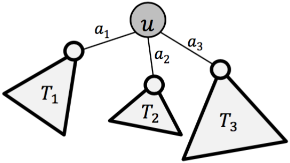
- 분할 정복을 합니다.
- Centroid 정점을 찾고, 이 정점을 지나는 모든 경로를 따져봅니다.
- 그 이후, Centroid를 잘라서 생기는 모든 부트리에 대해서 똑같은 문제를 재귀적으로 풉니다.
- 분할 정복을 할 때 $O(N)$ 에 한다면 시간 복잡도가 $O(N lg N)$ 이 됩니다.

Centroid Decomposition



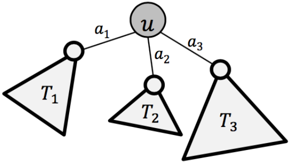
그래서 Centroid u를 찾았습니다. 이제 u를 지나는 모든 경로를 고려해 봅시다. 그걸 고려하는 건 조금 더 쉬울까요?

Centroid Decomposition



- 서로 다른 서브트리 T_i 에서 유래한 경로들을 봐줘야 합니다. 어떠한 정점 v가 속하는 서브트리 그룹을 G_v 라고 합시다. u 에서 다른 정점 i까지의 원소의 합을 S_i 라고 하면
- $S_i + S_j = K + S_u$ and $G_i \neq G_j$ 를 만족하는 쌍 i, j를 세면 됩니다.

Centroid Decomposition



- $S_i + S_j = K + S_u$ and $G_i \neq G_j$ 를 만족하는 쌍 i, j를 세면 됩니다.
- 이 부분에 대해서는 자세한 설명을 생략하겠지만, 다행이도 크게 어렵지 않습니다.
- 결국 u를 루트로 한 dfs와 정렬로 $O(nlgn)$ 정도에 문제를 해결할 수 있고 전체 문제는 $O(nlg^2n)$에 해결 가능합니다.