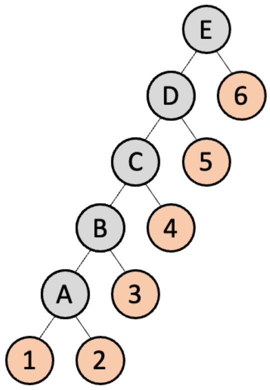
"중등부 3번 / 고등부 2번. 이진 트리" 문제 풀이

작성자: 박찬솔

부분문제 3

T_6의 형태는 아래와 같다.



노드 A, B, C, D, E가 덮는 리프 노드의 구간은 $[1, 2]$, $[1, 3]$, $[1, 4]$, $[1, 5]$, $[1, 6]$이다. 따라서, $f(1, i) = 1$이다.

$i \geq 2$인 경우, 번호가 i 이상 j 이하인 리프 노드를 덮으려면, 번호가 i 이상 j 이하인 리프 노드들을 모두 선택할 수밖에 없다. 따라서, $f(i, j) = j - i + 1$이다.

즉, $S(T_N) = \sum_{j=1}^{N} 1 + \sum_{i=2}^{N} \sum_{j=i}^{N} (j - i + 1)$ 이다. 이 값은 $O(1)$에 계산할 수 있다.

부분문제 4

모든 (a, b) 순서쌍에 대해 $f(a, b)$의 값을 직접 구하려고 한다.

주어진 트리를 깊이 우선 탐색 등의 방법으로 순회하면, 각 노드 u를 선택했을 때 덮이는 리프 노드 번호의 구간 $[l_u, r_u]$을 계산해 둘 수 있다.

$g(u, a, b)$를, "구간 $[a, b]$가 노드 u의 서브 트리에 포함됨을 알고 있을 때, $f(a, b)$의 값"이라고 정의하자. $f(a, b)$의 값은 $g(\texttt{root}, a, b)$ 함수의 값을 계산하면 구할 수 있다. g의 값을 재귀적으로 계산하여, $f(a, b)$의 값을 구할 것이다.

- $a = l_u$이고 $b = r_u$라면, 노드 u를 선택하면 구간 $[a, b]$가 덮이므로, $g(u, a, b) = 1$이다. 노드 u가 리프 노드였다면, 반드시 여기에 해당한다는 점에 유의하라.
- 노드 u의 왼쪽 자식을 v, 오른쪽 자식을 w라고 하자. $l_u = l_v$, $r_v + 1 = l_w$, $r_w = r_u$ 이다.
- 구간 $[a, b]$가 왼쪽 자식 v의 서브 트리에 포함된다면 (즉 $b \leq r_v$라면), $g(u, a, b) = g(v, a, b)$이다.
- 구간 $[a, b]$가 오른쪽 자식 w의 서브 트리에 포함된다면 (즉 $l_w \leq a$라면), $g(u, a, b) = g(w, a, b)$이다.
- 이제 $a \leq r_v < l_w \leq b$이다. 구간 $[a, r_v]$는 왼쪽 자식 v의 서브 트리에 포함되고, 구간 $[l_w, b]$는 오른쪽 자식 w의 서브 트리에 포함되며, 서로 겹치지 않으므로 독립적으로 계산할 수 있다. 즉, $g(u, a, b) = g(v, a, r_v) + g(w, l_w, b)$이다.

부분문제 1

위와 같은 g의 정의는, 완전 이진 트리 형태로 표현된 세그먼트 트리에서, 닫힌 구간 $[a,b]$의 정보를 추출하기 위해 세그먼트 트리를 순회하는 함수와 그 형태가 유사하다. $f(a,b)$의 값은 세그먼트 트리에서 구간 $[a,b]$에 대한 정보를 계산하기 위해, 값을 가져와야 하는 노드의 최소 개수라고 생각할 수 있다.

해당 함수를 사용하여 $O(\log |T_N|)$ 시간에 $f(a,b)$를 구할 수 있다. 모든 (a,b) 순서쌍에 대해 $f(a,b)$의 값을 구한 후 합쳐주면 된다.

부분문제 5

부분문제 4의 풀이는 모든 순서쌍 (a,b)에 대해, 구간 $[a,b]$를 덮기 위해 최소한으로 선택해야 하는 노드들을 직접 구한다고 볼 수 있다.

관점을 바꿔서, 각각의 노드에 대해, 위의 절차에서 그 노드를 선택하는 순서쌍 (a,b)의 개수가 몇 개인지를 세어보자.

부분 문제 4에서와 같이, 트리 전체를 순회하여, 각 노드 u를 선택했을 때 덮이는 리프 노드 번호의 구간 $[l_u, r_u]$을 계산해 두었다고 하자.

노드 u의 부모 노드가 p일 때, 구간 $[a,b]$를 덮을 때 노드 u가 선택되는 조건은, 아래 두 조건과 같다.

- 노드 u를 선택했을 때 구간 $[a,b]$ 바깥의 리프 노드가 덮이면 안 된다. 즉, $a \le l_u \le r_u \le b$여야 한다.
- 노드 p를 선택했을 때, 반드시 구간 $[a,b]$ 바깥의 리프 노드 하나 이상은 덮여야 한다. 만약 그렇지 않다면, u 대신 p를 선택하면 되기 때문이다. 즉, $l_p < a$ 이거나 $b < r_v$ 여야 한다.

위의 두 조건을 동시에 만족하는 (a,b)의 범위는 직사각형 형태로 나타나며, 그 개수를 $O(1)$ 시간에 구할 수 있다.

모든 노드 u에 대해 위와 같이 (a,b)의 개수를 세어 준 후 합쳐 주면, $S(T)$를 $\mathcal{O}(T\text{의 노드 개수})$ 시간에 계산할 수 있다.

부분문제 6

부분문제 2와 같이 완전 이진 트리가 주어지는 경우 리프 노드 개수가 2^N개로 상당히 많으므로, 모든 노드를 순회하는 방식으로는 계산할 수 없다.

T의 루트 노드가 u, u의 왼쪽 자식 노드가 v, 오른쪽 자식 노드가 w라고 하자. $f(a,b)$의 값을 제시된 대로 부분문제 4에서 제시된 것과 같이 여러 경우를 나누어서 계산해 보자.

$$\begin{aligned} S(T_u) &= \sum_{l_u \le a \le b \le r_u} f(a,b) \\ &= \sum_{l_v \le a \le b \le r_v} f(a,b) + \sum_{l_w \le a \le b \le r_w} f(a,b) + \sum_{l_v \le a \le r_v < l_w \le b \le r_w} \{f(a, r_v) + f(l_w, b)\} - 1 \end{aligned}$$

마지막에 1을 빼 주는 이유는, $a = l_u$, $b = r_u$일 때 $f(a,b) = 1$인 케이스를 간단하게 처리해 주기 위함이다.

$\sum_{l_v \le a \le b \le r_v} f(a,b)$는 v의 서브 트리 내의 모든 (a,b) 순서쌍에 대한 $f(a,b)$의 합이므로, $S(T_v)$와 같다.

$\sum_{l_w \le a \le b \le r_w} f(a,b)$는 w의 서브 트리 내의 모든 (a,b) 순서쌍에 대한 $f(a,b)$의 합이므로, $S(T_w)$와 같다.

이제 $\sum_{l_v \le a \le r_v < l_w \le b \le r_w} \{f(a, r_v) + f(l_w, b)\}$의 값을 계산하자. $f(a, r_v)$에는 b가 나타나지 않고, $f(l_w, b)$에는 a가 나타나지 않는다는 점에서 착안하여, 이 식을 전개하면 아래와 같다.

$$\begin{aligned}
&\sum_{l_v \le a \le r_v < l_w \le b \le r_w} \{f(a, r_v) + f(l_w, b)\} \\
=&\sum_{l_v \le a \le r_v < l_w \le b \le r_w} f(a, r_v) + \sum_{l_v \le a \le r_v < l_w \le b \le r_w} f(l_w, b) \\
=&(r_w - l_w + 1) \sum_{l_v \le a \le r_v} f(a, r_v) + (r_v - l_v + 1) \sum_{l_w \le b \le r_w} f(l_w, b)
\end{aligned}$$

$r_w - l_w + 1$는 w의 서브 트리 내의 리프 노드 개수이다. $r_v - l_v + 1$는 v의 서브 트리 내의 리프 노드 개수이다. 노드 u의 서브 트리 내의 리프 노드 개수를 $\text{cnt}[u]$라고 정의하자.

$\sum_{l_v \le a \le r_v} f(a, r_v)$는 v의 서브 트리에 있는 모든 리프 노드 a에 대해, $f(a, r_v)$의 합이다. 이 값을 $\text{suffix}[v]$라고 정의하자.

$\sum_{l_w \le b \le r_w} f(l_w, b)$는 w의 서브 트리에 있는 모든 리프 노드 b에 대해, $f(l_w, b)$의 합이다. 이 값을 $\text{prefix}[w]$라고 정의하자.

이 값들을 관리할 수 있다면, $S(T_i) = S(T_{A_i}) + S(T_{B_i}) + \text{suffix}[A_i] \times \text{cnt}[B_i] + \text{prefix}[B_i] \times \text{cnt}[A_i] - 1$으로 계산할 수 있다.

$\text{cnt}[0] = 1$이고, $\text{cnt}[i] = \text{cnt}[A_i] + \text{cnt}[B_i]$이다.

$\text{prefix}[i] = \text{prefix}[A_i] + (\text{prefix}[B_i] + cnt[B_i] - 1)$ 이다. 왼쪽 서브 트리의 f 값들은 그대로 유지되고, 오른쪽 서브 트리의 f 값들은 1 증가하며, 루트 노드의 f 값은 1이기 때문이다.

$\text{suffix}[i] = (\text{suffix}[A_i] + cnt[A_i] - 1) + \text{suffix}[B_i]$ 이다. 오른쪽 서브 트리의 f 값들은 그대로 유지되고, 왼쪽 서브 트리의 f 값들은 1 증가하며, 루트 노드의 f 값은 1이기 때문이다.

따라서 필요한 모든 값들을 상수 시간에 계산할 수 있으며, 문제를 $O(N)$에 해결할 수 있다.