Sqrt Decomposition

- 구간 쿼리 문제들을 처리하는 자료구조(혹은 method)의 일종.
- 하나의 쿼리를 처리하는데 $O(\sqrt{n})$의 시간복잡도를 가짐.
- 주어진 크기 n의 구간을 $O(\sqrt{n})$ 크기의 block들로 나눔.
  전체 block의 개수도 $O(\sqrt{n})$.
  - 구체적으로 $\lfloor \sqrt{n} \rfloor$, $\lceil \sqrt{n} \rceil$ 등 다양한 기준이 가능.
- block의 크기를 s라 할 때, 다음과 같은 형태.

$$\underbrace{a[0], a[1], \dots, a[s-1]}_{b[0]}, \underbrace{a[s], a[s+1], \dots, a[2s-1]}_{b[1]},$$

$$\dots, \underbrace{a[(s-1)\cdot s], \dots, a[n-1]}_{b[s-1]}.$$

※ 마지막 block $b[s-1]$의 크기는 s가 아닐 수 있음(n이 s의 배수가 아닐 때).

- 구간합을 관리한다고 할 때

$$b[k] = \sum_{i=k\cdot s}^{\min(n-1,\,(k+1)\cdot s-1)} a[i]$$

• 구간 $[l, r]$의 합을 구하는 쿼리가 주어질 때,
원소 $p \in [l, r]$은 다음 중 하나에 속한다.

① 어떤 k에 대해
$p \in [k \cdot s, (k+1) \cdot s - 1] \subset [l, r]$를 만족.
(i.e., p는 구간에 속하는 어떤 block $b[k]$의 원소.)

② p는 ①에 해당하지 않음.
(i.e., p는 구간 $[l, r]$의 '양 끝')

block에 속하는 원소들은 block 노드로 처리한다.
block에 속하지 않는 원소들은 각각 처리한다.

block의 개수는 $O(\sqrt{n})$개, block에 속하지 않는
원소의 개수도 $O(\sqrt{n})$개 이므로,
쿼리 하나를 처리하는 시간복잡도는 $O(\sqrt{n}) + O(\sqrt{n}) = O(\sqrt{n})$이다.

• sqrt decomposition은 구간합 연산 외에도
다양한 구간 연산에 적용될 수 있다.
(e.g., finding the number of zero elements, finding the first non-zero element, counting elements which satisfy a certain property etc)

Mo's Algorithm

- 많은 구간 쿼리를 빠르게 처리할 수 있는 오프라인 쿼리 알고리즘의 일종.

- 원소의 수정이 없고, 구간 내에서 어떤 결과를 찾는 종류의 쿼리에만 적용 가능.

- 쿼리 Q_i를 '구간 $[s_i, e_i]$의 구간 합을 출력하라'라고 정의하자.
  쿼리들을 다음 기준으로 정렬하자.
  ① $\lfloor s_i / \sqrt{n} \rfloor < \lfloor s_j / \sqrt{n} \rfloor$ → 꼭 floor function을 필요는 X. 핵심은 $\sqrt{n}$이 같은 값들을 동일하게 취급한다는 것.
  ② $e_i < e_j$

- 위와 같이 쿼리들을 정렬한 후에 Q_{i+1}을 구할 때 Q_i에서 구한 값을 재사용한다.
  따라서 우리는 구간 $[s_i, e_i] \cap [s_{i+1}, e_{i+1}]$의 값은 재사용하고, $([s_i, e_i] \cup [s_{i+1}, e_{i+1}]) - ([s_i, e_i] \cap [s_{i+1}, e_{i+1}])$은 새로 계산해야 한다.
  이 과정에서 총 $O(|s_{i+1} - s_i| + |e_{i+1} - e_i|)$번 계산해야 한다.

◦ 모든 재사용 과정의 시간복잡도의 합을 계산해보자. (m은 쿼리의 총 개수)

① 바로 앞선 쿼리와 $\lfloor S/\sqrt{n} \rfloor$이 같을 경우

$|S_{i+1} - S_i| = O(\sqrt{n})$이고 쿼리의 개수가 $O(m)$이므로

시작점은 $O(m\sqrt{n})$번 이동.

끝점 e는 단조 증가하므로 끝점은 총 $O(n)$번 바뀌고,

block의 총 $O(\sqrt{n})$개 이므로 $O(n\sqrt{n})$번 이동.

② 바로 앞선 쿼리와 $\lfloor S/\sqrt{n} \rfloor$이 다를 경우

$|S_{i+1} - S_i| = O(n)$이고, ②번 case 과정이 $O(\sqrt{n})$번

일어남. $O(n\sqrt{n})$번 이동.

끝점의 경우에도 마찬가지로 $|e_{i+1} - e_i| = O(n)$이고

②번 case가 $O(\sqrt{n})$번 일어나므로 $O(n\sqrt{n})$번 이동.

∴ 전체 이동의 횟수는 $O((n+m)\sqrt{n})$이다.

시간복잡도는 $O((n+m)\sqrt{n}) \times$ (이동 연산의 시간복잡도)