

熱帯雨林における跳躍 (Rainforest Jumps)

スマトラの熱帯雨林には N 本の樹が横一列に並んでおり、左から順に 0 から $N - 1$ の番号が付けられている。樹 i の高さは $H[i]$ であり、これらは互いに相異なる。

Pak Dengklek は、オランウータンが樹々の間を飛び移れるように訓練している。1 回の飛び移りに際して、オランウータンは今いる樹の頂上から、今いる樹より高い樹の中で左右いずれかに最も近い樹の頂上に飛び移ることができる。厳密に表現すると、以下のいずれかを満たすとき、またそのときに限り、オランウータンは樹 x から樹 y に飛び移ることができる。

- y は x より小さい非負整数のうち、 $H[y] > H[x]$ を満たす最大のものである。
- y は x より大きい非負整数のうち、 $H[y] > H[x]$ を満たす最小のものである。

Pak Dengklek は Q 個のプランを考えており、各プランは 4 つの整数 A, B, C, D ($A \leq B < C \leq D$) で表される。Pak Dengklek は各プランについて、オランウータンが飛び移りを何回か繰り返すことで、樹 $A, A + 1, \dots, B$ のいずれかから出発して樹 $C, C + 1, \dots, D$ のいずれかにたどり着くことができるかどうかを知りたい。もし達成可能であれば、そのために飛び移る回数の最小値も求めてほしい。

実装の詳細 (Implementation Details)

以下の関数を実装せよ。

```
void init(int N, int[] H)
```

- N : 樹の本数。
- H : 長さ N の配列であり、 $H[i]$ は樹 i の高さを表す。
- この関数は 1 回だけ呼び出される。この関数が呼び出される前に `minimum_jumps` が呼び出されることはない。

```
int minimum_jumps(int A, int B, int C, int D)
```

- A, B : オランウータンが飛び移りを始められる樹の範囲。
- C, D : オランウータンが飛び移りを終わられる樹の範囲。
- この関数は、プランが達成可能ならば必要な飛び移りの回数の最小値を、プランが達成不可能ならば -1 を戻り値として返すこと。
- この関数はちょうど Q 回呼び出される。

例 (Example)

以下の関数呼び出しを考える。

```
init(7, [3, 2, 1, 6, 4, 5, 7])
```

初期設定が完了したら、以下の関数呼び出しを考える。

```
minimum_jumps(4, 4, 6, 6)
```

これは、オランウータンが樹 4 (高さ 4) から出発して樹 6 (高さ 7) に到着するというプランを意味する。最小の飛び移り回数で達成する方法の一例として、まず樹 3 (高さ 6) に、そのあと樹 6 に飛び移るというものがある。あるいは、まず樹 5 (高さ 5) に、そのあと樹 6 に飛び移るという方法も考えられる。したがって、関数 `minimum_jumps` は 2 を返さなければならない。

次に、以下の関数呼び出しを考える。

```
minimum_jumps(1, 3, 5, 6)
```

これは、オランウータンが樹 1 (高さ 2), 樹 2 (高さ 1), 樹 3 (高さ 6) のいずれかから出発して、樹 5 (高さ 5), 樹 6 (高さ 7) のいずれかに到着するというプランを意味する。最小の飛び移り回数で達成する唯一の方法は、樹 3 から出発して、樹 6 に直接飛び移るというものである。したがって、関数 `minimum_jumps` は 1 を返さなければならない。

次に、以下の関数呼び出しを考える。

```
minimum_jumps(0, 1, 2, 2)
```

これは、オランウータンが樹 0 (高さ 3), 樹 1 (高さ 2) のいずれかから出発して、樹 2 (高さ 1) に到着するというプランを意味する。樹 2 は最も低い樹であるため、より高い他の樹からたどり着くことは不可能である。したがって、関数 `minimum_jumps` は -1 を返さなければならない。

制約 (Constraints)

- $2 \leq N \leq 200\,000$.
- $1 \leq Q \leq 100\,000$.
- $1 \leq H[i] \leq N$ ($0 \leq i \leq N - 1$).
- $H[i] \neq H[j]$ ($0 \leq i < j \leq N - 1$).
- $0 \leq A \leq B < C \leq D \leq N - 1$.

小課題 (Subtasks)

1. (4 点) $H[i] = i + 1$ ($0 \leq i \leq N - 1$).
2. (8 点) $N \leq 200, Q \leq 200$.

3. (13 点) $N \leq 2000, Q \leq 2000$.

4. (12 点) $Q \leq 5$.

5. (23 点) $A = B, C = D$.

6. (21 点) $C = D$.

7. (19 点) 追加の制約はない.

採点プログラムのサンプル (Sample Grader)

採点プログラムのサンプルは入力データを以下の形式で読み込む.

- 1 行目: $N Q$
- 2 行目: $H[0] H[1] \dots H[N - 1]$
- $3 + i$ 行目 ($0 \leq i \leq Q - 1$): $A B C D$ (関数 `minimum_jumps` の i 番目の呼び出しの引数)

採点プログラムのサンプルはあなたの解答を以下の形式で出力する.

- $1 + i$ 行目 ($0 \leq i \leq Q - 1$): 関数 `minimum_jumps` の i 番目の呼び出しにおける戻り値.