Věžák

V alternativním vesmíru je Vlad uvězněn uvnitř futuristické verze pevnosti Poenari skládající se z n patrech očíslovaných od 0 do $n-1$. Z každého patra i ($0 \leq i \leq n-1$) může jít pouze nahoru, buď půjde po schodech začínajících na daném patře a zaplatí 1 kapku krve (to je měna, kterou upíři a podobná verbež používají k placení v Rumunsku), nebo se promění v netopýra a proletí větracími otvory vycházející z daného patra, za což musí zaplatit 2 kapky krve. Schody ho mohou vynést až o $v[i]$ pater nahoru, zatímco větrací otvory až o $w[i]$ pater nahoru, kde v a w jsou dvě zadaná pole: $v = v[0], v[1], \ldots, v[n-1]$ a $w = w[0], w[1], \ldots, w[n-1]$.

Přesněji řečeno: Z patra i ($0 \leq i \leq n-1$) se Vlad může přesunout:

- kamkoliv od patra $i+1$ do patra $i+v[i]$ (včetně), pokud nepřekročí patro $n-1$, za cenu 1 kapky krve,
- kamkoliv od patra $i+1$ do patra $i+w[i]$ (včetně), pokud nepřekročí patro $n-1$, za cenu 2 kapek krve.

  Vladův bratr Miroslav mu navrhl m scénářů, každý z nich složený z dvojice pater A a B ($A \leq B$). Zajímalo by ho, kolik nejméně kapek krve Vlad musí zaplatit, pokud by se chtěl dostat z patra A do patra B.

Implementační podrobnosti

Máte implementovat funkci `solve` následující signatury:

```
std::vector<int> solve(std::vector<int> &v, std::vector<int> &w,
    std::vector<std::pair<int,int>> &queries);
```

- Funkce obdrží vektor v – výšky schodišť z jednotlivých pater a w – výšky větracích systémů, oba délky n.
- Dále funkce obdrží scénáře (dotazy) jako vektor dvojic délky m. Každá z nich obsahuje čísla pater A a B, jak bylo popsáno v zadání.
- Funkce má vrátit vektor délky m, obsahující odpovědi na všech m dotazů.

Omezení

- $1 \leq n, m \leq 500\,000$.
- $1 \leq v[i], w[i] \leq n$ pro každé $0 \leq i \leq n-1$.
- $0 \leq A \leq B \leq n-1$ pro každý scénář.

Podúlohy

1. (5 bodů) $1 \leq n \leq 300,\ 1 \leq m \leq 500\,000$
2. (7 bodů) $1 \leq n \leq 3\,000,\ 1 \leq m \leq 3\,000$
3. (11 bodů) $1 \leq n \leq 20\,000,\ 1 \leq m \leq 20\,000$
4. (44 bodů) $1 \leq n \leq 200\,000,\ 1 \leq m \leq 200\,000$
5. (8 bodů) $1 \leq n \leq 500\,000,\ 1 \leq m \leq 500\,000,\ \ v[i] \leq v[j]$ a $w[i] \leq w[j]$ pro každé $0 \leq i < j \leq n-1$
6. (25 bodů) *Bez dalších omezení.*

Příklady

Příklad 1

Uvažme následující volání:

```
solve({2, 3, 1, 1, 1, 1, 2}, {3, 4, 1, 2, 1, 2, 2},
    {{0, 4}, {0, 5}, {0, 6}})
```

Zde máme $n = 7$ a 3 dotazy, $v = [2,3,1,1,1,1,2]$ a $w = [3,4,1,2,1,2,2]$.

V prvním scénáři $(0, 4)$ může Vlad udělat dva přesuny o ceně 1: z 0 do 1 (mohl by sice za stejnou cenu až do patra 2, ale z patra 1 se dostane dál), a pak z 1 fo 4. Celková cena: $1 + 1 = 2$.

Ve druhém scénáři $(0, 5)$ existují dvě optimální cesty: z 0 do 1 (cena 1), z 1 do 4 (cena 1) z 4 do 5 (cena 1); druhá možnost je z 0 do 1 (cena 1), z 1 do 5 (cena 2). Celková cena: $1 + 1 + 1 = 1 + 2 = 3$.

Pro třetí scénář $(0, 6)$ si ukážeme jednu ukázkovou cestu ceny 4: z 0 do 1 (cena 1), 1 to 5 (cena 2), 5 to 6 (cena 1). Celková cena: $1 + 2 + 1 = 4$

Funkce tedy musí vrátit:

```
{2, 3, 4}
```

Příklad 2

A teď uvažme toto zavolání:

```
solve({1, 1, 1, 2, 3, 2, 1, 1, 2, 3}, {2, 4, 1, 4, 1, 4, 1, 3, 2, 3},
    {{3, 9}, {0, 9}, {0, 7}, {0, 4}, {3, 5}})
```

Zde jsou optimální cesty:

- $(3{,}9)$: z 3 do 5 (cena 1), z 5 do 9 (cena 2) $\Longrightarrow$ celkem: 3
- $(0{,}9)$: z 0 do 1 (cena 1), z 1 do 5 (cena 2), z 5 do 9 (cena 2) $\Longrightarrow$ celkem: 5
- $(0{,}7)$: z 0 do 1 (cena 1), z 1 do 5 (cena 2), z 5 do 7 (cena 1) $\Longrightarrow$ celkem: 4
- $(0{,}4)$: z 0 do 1 (cena 1), z 1 do 4 (cena 2) $\Longrightarrow$ celkem: 3
- $(3{,}5)$: z 3 do 5 (cena 1) $\Longrightarrow$ celkem: 1

Funkce tedy musí vrátit:

```
{3, 5, 4, 3, 1}
```

Ukázkový grader

Ukázkový grader čte vstup v následujícím formátu:

- řádek 1: n
- řádek 2: $v[0]\ v[1] \dots v[n-1]$
- řádek 3: $w[0]\ w[1] \dots w[n-1]$
- řádek 4: m
- řádek $5+i (0 \le i \le m-1)$: $A\ B$

a vypíše m řádků – výstup `solve`.