Najekonomickejšie narábanie s krvou s cieľom presnej prepravy medzi dvoma konkrétnymi bodmi s využitím dvoch módov transportu

V alternatívnom vesmíre je Vlad uväznený vo futuristickej verzii hradu Poenari, ktorý teraz pozostáva z n poschodí, očíslovaných od 0 po $n-1$. Z každého poschodia i ($0 \leq i \leq n-1$) sa môže pohybovať iba nahor – buď po schodoch za cenu 1 kvapky krvi (mena, ktorou upíri platia v Rumunsku), alebo sa môže premeniť na netopiera a preletieť cez ventilačné šachty, čo ho stojí 2 kvapky krvi. Schody ho môžu vyniesť najviac o $v[i]$ poschodí vyššie, ventilačné šachty až o $w[i]$ poschodí vyššie, kde v a w sú dané polia:
$v = v[0], v[1], \ldots, v[n-1]$
$w = w[0], w[1], \ldots, w[n-1]$.

Formálne, z poschodia i ($0 \leq i \leq n-1$), Vlad môže ísť:

- na akékoľvek poschodie od $i+1$ do $\min(n-1, i+v[i])$, za cenu 1
- na akékoľvek poschodie od $i+1$ do $\min(n-1, i+w[i])$, za cenu 2

Ďalej jeho bratia Radu a Mircea navrhli m scenárov, pričom každý pozostáva z dvoch poschodí A a B ($A \leq B$). Vlad musí odpovedať na ich m otázok: aké je **minimálne množstvo krvi**, ktoré musí obetovať, aby sa dostal z poschodia A na poschodie B?

Implementačné detaily

Treba implementovať funkciu `solve`:

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

- Prijíma polia v (výšky schodov) a w (výšky ventilačných šácht), obe veľkosti n.
- Tiež prijíma pole `queries`, vektor dvojíc veľkosti m, každá dvojica obsahuje A a B ako bolo popísané v zadaní.
- Funkcia vracia vektor veľkosti m s odpoveďami na m otázok.

Obmedzenia

- $1 \leq n, m \leq 500\,000$
- $1 \leq v[i], w[i] \leq n$ pre všetky $0 \leq i \leq n-1$
- $0 \leq A \leq B \leq n-1$ pre všetky otázky

Podúlohy

1. (5 bodov) $1 \leq n \leq 300,\ 1 \leq m \leq 500\,000$
2. (7 bodov) $1 \leq n \leq 3\,000,\ 1 \leq m \leq 3\,000$
3. (11 bodov) $1 \leq n \leq 20\,000,\ 1 \leq m \leq 20\,000$
4. (44 bodov) $1 \leq n \leq 200\,000,\ 1 \leq m \leq 200\,000$
5. (8 bodov) $1 \leq n \leq 500\,000,\ 1 \leq m \leq 500\,000,$ kde $v[i] \leq v[j]$ a $w[i] \leq w[j]$ pre všetky $0 \leq i < j \leq n-1$
6. (25 bodov) Bez ďalších obmedzení

Príklady

Príklad 1

Pozrime sa na nasledujúce volanie:

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

Máme $n = 7$ a $m = 3$ otázky, $v = [2, 3, 1, 1, 1, 1, 2]$ a $w = [3, 4, 1, 2, 1, 2, 2]$.

Pre prvú otázku $(0, 4)$ musí Vlad urobiť dva 1-kvapkové skoky: z 0 na 1 (aj keď môže skočiť na 2, poschodie 1 ho dostane ďalej), potom z 1 na 4. Celková cena: $1 + 1 = 2$.

Pre druhú otázku $(0, 5)$ existujú 2 optimálne cesty: 0 na 1 (cena 1), 1 na 4 (cena 1), 4 na 5 (cena 1); druhá cesta je 0 na 1 (cena 1), 1 na 5 (cena 2). Celková cena: $1 + 1 + 1 = 1 + 2 = 3$.

Pre tretiu otázku $(0, 6)$, príklad jednej cesty s cenou 4 je: 0 na 1 (cena 1), 1 na 5 (cena 2), 5 na 6 (cena 1). Celková cena: $1 + 2 + 1 = 4$

Takže vector, ktorý funkcia vráti, musí byť:

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

Príklad 2

Pozrime sa na nasledujúce volanie:

```
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}}
)
```

Toto sú optimálne cesty pre jednotlivé otázky:

(3,9): 3 na 5 (cena 1), 5 na 9 (cena 2) $\implies$ dokopy: 3

(0,9): 0 na 1 (cena 1), 1 na 5 (cena 2), 5 na 9 (cena 2) $\implies$ dokopy: 5

(0,7): 0 na 1 (cena 1), 1 na 5 (cena 2), 5 na 7 (cena 1) $\implies$ dokopy: 4

(0,4): 0 na 1 (cena 1), 1 na 4 (cena 2) $\Longrightarrow$ dokopy: 3

(3,5): 3 na 5 (cena 1) $\Longrightarrow$ dokopy: 1

Takže vector, ktorý funkcia vráti, musí byť:

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

Hodnotič príkladov

Hodnotič príkladov číta vstup v nasledujúcom formáte:

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

a vypíše m riadkov — výsledky volania `solve`.