

Найвище

В альтернативному всесвіті Степан застряг у футуристичній версії фортеці Поенарі, яка має n поверхів, пронумерованих від 0 до $n - 1$. З кожного поверху i ($0 \leq i \leq n - 1$), він може піднятися лише вгору, або піднявшись сходами та заплативши 1 (це валюта, яку використовують вампіри в Румунії), або перетворившись на кажана і пролетівши через вентиляцію, що коштує 2 краплі крові. Сходами він може піднятися максимум на $v[i]$ поверхів вгору, а вентиляцією — максимум на $w[i]$ поверхів вгору, де v та w — це два задані масиви: $v = v[0], v[1], \dots, v[n - 1]$ та $w = w[0], w[1], \dots, w[n - 1]$.

Формально, з поверху i ($0 \leq i \leq n - 1$), Степан може перейти:

- на будь-який поверх від $i + 1$ до $i + v[i]$ не перевищуючи $n - 1$, за 1 краплю крові;
- на будь-який поверх від $i + 1$ до $i + w[i]$ не перевищуючи $n - 1$, за 2 краплі крові.

Крім того, його брати Валера та Ілля запропонували m сценаріїв для Степана, кожен з яких складається з двох поверхів A та B ($A \leq B$). Степан має відповісти на m запитань: яку мінімальну кількість крові він має пожертвувати, щоб дістатися з поверху A на поверх B ?

Деталі реалізації

Вам потрібно буде реалізувати функцію solve:

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

- Отримує вектори v — висоти маршів сходів — та w — висоти вентиляційних систем, що починаються з кожного поверху. Обидва вектори мають розмір n .
- Також отримує запити — вектор пар розміром m . Кожна пара містить значення A та B , як описано в умові задачі.
- Повертає вектор розміру m , що складається з відповідей на m запитів.

Обмеження

- $1 \leq n, m \leq 500\,000$.
- $1 \leq v[i], w[i] \leq n$ для всіх $0 \leq i \leq n - 1$.
- $0 \leq A \leq B \leq n - 1$ для всіх запитів.

Підзадачі

1. (5 points) $1 \leq n \leq 300$, $1 \leq m \leq 500\,000$
2. (7 points) $1 \leq n \leq 3\,000$, $1 \leq m \leq 3\,000$
3. (11 points) $1 \leq n \leq 20\,000$, $1 \leq m \leq 20\,000$
4. (44 points) $1 \leq n \leq 200\,000$, $1 \leq m \leq 200\,000$
5. (8 points) $1 \leq n \leq 500\,000$, $1 \leq m \leq 500\,000$, $v[i] \leq v[j]$ та $w[i] \leq w[j]$ для всіх $1 \leq i \leq j \leq n$
6. (25 points) Ніяких додаткових обмежень.

Приклади

Приклад 1

Розглянемо наступний виклик:

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

Тут ми маємо $n = 7$ та 3 запити, $v = [2, 3, 1, 1, 1, 1, 2]$ та $w = [3, 4, 1, 2, 1, 2, 2]$.

Для першого запиту (0, 4), Степан має зробити два стрибки вартістю 1 краплю крові: 0 до 1 (хоча він може стрибнути до 2, поверх 1 дозволить йому пройти далі), потім з 1 до 4. Загальна вартість: $1 + 1 = 2$.

Для другого запиту (0, 5), існує 2 оптимальні шляхи: 0 до 1 (вартість 1), 1 до 4 (вартість 1), 4 до 5 (вартість 1); другий шлях з 0 до 1 (вартість 1), 1 до 5 (вартість 2). Загальна вартість: $1 + 1 + 1 = 1 + 2 = 3$.

Для третього запиту $(0, 6)$, приклад шляху з вартістю 4 є 0 до 1 (вартість 1), 1 до 5 (вартість 2), 5 до 6 (вартість 1). Загальна вартість: $1 + 2 + 1 = 4$

Отже, вектор, який має повернути функція, буде таким:

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

Приклад 2

Розглянемо наступний виклик:

```
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}})
```

Це оптимальні шляхи для запитів:

$(3, 9)$: 3 до 5 (вартість 1), 5 до 9 (вартість 2) \implies Всього: 3

$(0, 9)$: 0 до 1 (вартість 1), 1 до 5 (вартість 2), 5 до 9 (вартість 2) \implies Всього: 5

$(0, 7)$: 0 до 1 (вартість 1), 1 до 5 (вартість 2), 5 до 7 (вартість 1) \implies Всього: 4

$(0, 4)$: 0 до 1 (вартість 1), 1 до 4 (вартість 2) \implies Всього: 3

$(3, 5)$: 3 до 5 (вартість 1) \implies Всього: 1

Отже, вектор, який має повернути функція, буде таким:

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

Приклад градера

Зчитує вхідні дані у такому форматі:

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

і виводить m рядків — результат виклику функції `solve`.