



Closing Time

Hungary là đất nước có N thành phố, đánh số từ 0 đến $N - 1$.

Các thành phố được nối với nhau bằng $N - 1$ con đường *hai chiều*, được đánh số từ 0 đến $N - 2$. Với mỗi j mà $0 \leq j \leq N - 2$, con đường đánh số j nối thành phố $U[j]$ với thành phố $V[j]$ có độ dài $W[j]$, nghĩa là, một người đi giữa hai thành phố mất $W[j]$ đơn vị thời gian. Mỗi con đường nối hai thành phố phân biệt và mỗi cặp thành phố được nối bởi nhiều nhất một con đường.

Một **đường đi** giữa hai thành phố phân biệt a và b là một chuỗi p_0, p_1, \dots, p_t các thành phố khác nhau, sao cho:

- $p_0 = a$,
- $p_t = b$,
- với mỗi i ($0 \leq i < t$), có một con đường nối thành phố p_i và p_{i+1} .

Có thể đi lại từ một thành phố bất kì đến một thành phố bất kì khác sử dụng các con đường, nghĩa là, có một đường đi giữa hai thành phố phân biệt bất kì. Có thể chứng minh rằng đường đi này là duy nhất cho từng cặp thành phố phân biệt.

Độ dài của một đường đi p_0, p_1, \dots, p_t là tổng độ dài của t con đường nối các thành phố liên tiếp nhau dọc theo đường đi đó.

Ở Hungary, nhiều người di chuyển để tham gia các lễ hội ngày Quốc khánh tại hai thành phố lớn. Khi các buổi ăn mừng kết thúc, họ quay trở về nhà của họ. Chính quyền muốn tránh việc các đám đông làm phiền cư dân địa phương, nên họ dự định ra lệnh giới nghiêm ở tất cả thành phố vào các thời điểm nhất định. Mỗi thành phố được gán một số không âm **thời điểm đóng cửa** bởi chính quyền. Chính quyền quyết định rằng tổng các thời điểm đóng cửa không vượt quá K . Cụ thể, với mỗi i nằm trong khoảng giữa 0 và $N - 1$ (kể cả 0 và $N - 1$), thời điểm đóng cửa được gán cho thành phố i là số nguyên không âm $c[i]$. Tổng của tất cả các số $c[i]$ không vượt quá K .

Xét thành phố a và một cách gán thời điểm đóng cửa. Ta nói thành phố b **đến được** từ thành phố a khi và chỉ khi hoặc $b = a$, hoặc đường đi p_0, \dots, p_t giữa hai thành phố này (nghĩa là $p_0 = a$ và $p_t = b$) thoả mãn các điều kiện sau đây:

- độ dài đường đi p_0, p_1 không vượt quá $c[p_1]$, và
- độ dài đường đi p_0, p_1, p_2 không vượt quá $c[p_2]$, và
- ...
- độ dài đường đi $p_0, p_1, p_2, \dots, p_t$ không vượt quá $c[p_t]$.

Trong năm nay, hai địa điểm chính tổ chức lễ hội đặt tại thành phố X và thành phố Y . Với mỗi cách gán thời điểm đóng cửa, **điểm thuận tiện** được định nghĩa là tổng của hai số sau:

- Số lượng thành phố đến được từ thành phố X .
- Số lượng thành phố đến được từ thành phố Y .

Lưu ý rằng nếu một thành phố đến được từ thành phố X và cũng đến được từ thành phố Y thì thành phố đó được tính *hai lần* vào điểm thuận tiện.

Nhiệm vụ của bạn là tính toán điểm thuận tiện lớn nhất có thể đạt được bởi một cách gán thời điểm đóng cửa.

Chi tiết cài đặt

Bạn phải cài đặt hàm sau:

```
int max_score(int N, int X, int Y, int64 K, int[] U, int[] V, int[] W)
```

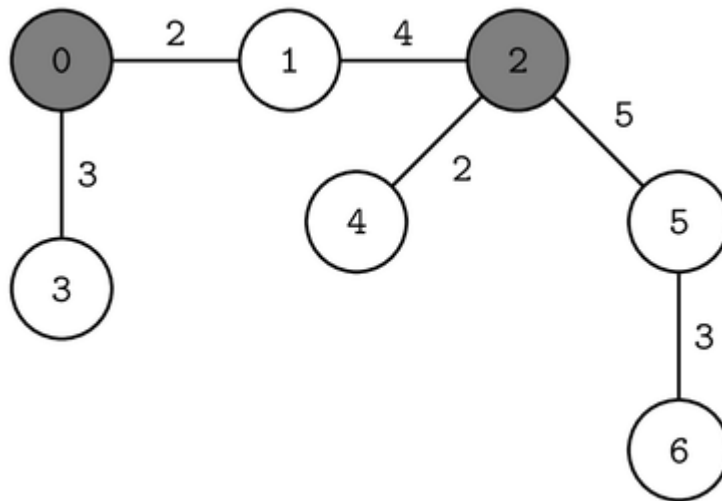
- N : số lượng thành phố.
- X, Y : hai thành phố có địa điểm chính tổ chức lễ hội.
- K : cận trên của tổng thời điểm đóng cửa.
- U, V : các mảng có kích thước $N - 1$ mô tả hai đầu của các con đường.
- W : mảng có kích thước $N - 1$ mô tả độ dài các con đường.
- Hàm này phải trả về điểm thuận tiện lớn nhất có thể đạt được bởi một cách gán thời điểm đóng cửa.
- Hàm này có thể được gọi **hiều lần** trong mỗi test.

Ví dụ

Xét lời gọi hàm sau:

```
max_score(7, 0, 2, 10,  
          [0, 0, 1, 2, 2, 5], [1, 3, 2, 4, 5, 6], [2, 3, 4, 2, 5, 3])
```

Lời gọi này tương ứng với mạng lưới các con đường sau:



Giả sử các thời điểm đóng cửa được gán như sau:

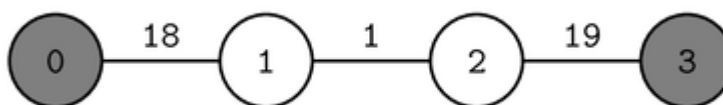
Thành phố	0	1	2	3	4	5	6
Thời điểm đóng cửa	0	4	0	3	2	0	0

Lưu ý rằng tổng thời điểm đóng cửa là 9, không vượt quá $K = 10$. Các thành phố 0, 1, và 3 có thể đến được từ thành phố X ($X = 0$), trong khi đó các thành phố 1, 2, and 4 có thể đến được từ thành phố Y ($Y = 2$). Vì thế, điểm thuận tiện là $3 + 3 = 6$. Không có cách gán thời điểm đóng cửa nào có điểm thuận tiện lớn hơn 6, do đó thủ tục trên phải trả về 6.

Xét lời gọi thủ tục sau:

```
max_score(4, 0, 3, 20, [0, 1, 2], [1, 2, 3], [18, 1, 19])
```

Lời gọi này tương ứng với mạng lưới các con đường sau:



Giả sử các thời điểm đóng cửa được gán như sau:

Thành phố	0	1	2	3
Thời điểm đóng cửa	0	1	19	0

Thành phố 0 có thể đến được từ thành phố X ($X = 0$), trong khi đó các thành phố 2 và 3 có thể đến được từ thành phố Y ($Y = 3$). Vì thế, điểm thuận tiện là $1 + 2 = 3$. Không có cách gán thời

điểm đóng cửa nào có điểm thuận tiện lớn hơn 3, do đó thủ tục trên phải trả về 3.

Các ràng buộc

- $2 \leq N \leq 200\,000$
- $0 \leq X < Y < N$
- $0 \leq K \leq 10^{18}$
- $0 \leq U[j] < V[j] < N$ (với mỗi j mà $0 \leq j \leq N - 2$)
- $1 \leq W[j] \leq 10^6$ (với mỗi j mà $0 \leq j \leq N - 2$)
- Có thể đi từ thành phố bất kì đến một thành phố bất kì khác sử dụng các con đường.
- $S_N \leq 200\,000$, với S_N là tổng của các giá trị N trong tất cả các lời gọi đến `max_score` trong mỗi test.

Các subtask

Ta nói một mạng lưới các con đường là **đường thẳng** nếu con đường i nối thành phố i và $i + 1$ (với mỗi i mà $0 \leq i \leq N - 2$).

1. (8 điểm) Độ dài đường đi từ thành phố X đến thành phố Y lớn hơn $2K$.
2. (9 điểm) $S_N \leq 50$, mạng lưới là đường thẳng.
3. (12 điểm) $S_N \leq 500$, mạng lưới là đường thẳng.
4. (14 điểm) $S_N \leq 3\,000$, mạng lưới là đường thẳng.
5. (9 điểm) $S_N \leq 20$
6. (11 điểm) $S_N \leq 100$
7. (10 điểm) $S_N \leq 500$
8. (10 điểm) $S_N \leq 3\,000$
9. (17 điểm) Không có ràng buộc nào thêm.

Trình chấm mẫu

Đặt C là số lượng kịch bản, tức là số lần gọi đến `max_score`. Trình chấm mẫu đọc dữ liệu vào theo định dạng sau:

- dòng 1: C

Mô tả của C kịch bản tiếp theo sau đó.

Trình chấm mẫu đọc mô tả của từng kịch bản theo định dạng sau:

- dòng 1: $N\ X\ Y\ K$
- dòng $2 + j$ ($0 \leq j \leq N - 2$): $U[j]\ V[j]\ W[j]$

Trình chấm mẫu in một dòng duy nhất cho từng kịch bản theo định dạng sau:

- dòng 1: giá trị trả về của `max_score`