



Closing Time

Հունգարիաում կա N հատ քաղաք համարակալված 0 -ից $N - 1$ թվերով:

Քաղաքներն իրար միացված են $N - 1$ հատ *երկկողմանի* ճանապարհներով, որոնք համարակալված են 0 -ից $N - 2$ թվերով: j -րդ ճանապարհը ($0 \leq j \leq N - 2$) իրար է միացնում $U[j]$ և $V[j]$ քաղաքները և ունի $W[j]$ երկարություն, դա նշանակում է, որ այդ ճանապարհը կարելի է անցնել $W[j]$ միավոր ժամանակում: Ամեն ճանապարհ միացնում է երկու տարբեր քաղաքներ և քաղաքների ցանկացած զույգ միացված է ամենաշատը մեկ ճանապարհով:

Երկու տարբեր a և b քաղաքների միջև **երթուղի** կանվանենք p_0, p_1, \dots, p_t տարբեր քաղաքների հաջորդականությունը այնպիսին որ.

- $p_0 = a$,
- $p_t = b$,
- ցանկացած i -ի համար ($0 \leq i < t$), գոյություն ունի ճանապարհ, որը միացնում է p_i և p_{i+1} քաղաքները:

Ցանկացած քաղաքից հնարավոր է հասնել մնացած բոլոր քաղաքներին, այսինքն ցանկացած երկու տարբեր քաղաքների միջև գոյություն ունի երթուղի: Նկատել, որ այդ երթուղին միակն է:

p_0, p_1, \dots, p_t երթուղու **երկարություն** կանվանենք դրա մեջ մտնող t ճանապարհների երկարությունների գումարը:

Հունգարիաում շատ մարդիկ ճամփորդում են անկախության օրվա տոնակատարություններին մասնակցելու համար: Կառավարությունը ուզում է միջոցառումներ ձեռնարկել, որպեսզի տեղացիներին անհարմարություններ չպատճառվեն. նրանք պլանավորում են փակել բոլոր քաղաքները ինչոր ժամանակներում: Ամեն քաղաքի համար ընտրվելու է **փակման ժամանակ**: Կառավարությունը որոշել է, որ փակման ժամանակների գումարը չպետք է գերազանցի K թիվը: Ավելի կոնկրետ, 0 -ից $N - 1$ ցանկացած i -ի համար, փակման ժամանակը ոչ բացասական ամբողջ է՝ $c[i]$: Բոլոր $c[i]$ -ների գումարը չպետք է գերազանցի K թիվը:

Դիտարկենք a քաղաքը և փակման ժամանակների ինչոր ընտրություն: Կասենք, որ b քաղաքը **հասանելի** է a քաղաքից, եթե $b = a$, կամ այդ երկու քաղաքները միացնող p_0, \dots, p_t երթուղին (պարզ է, որ $p_0 = a$ և $p_t = b$) բավարարում է հետևյալ պայմաններին.

- p_0, p_1 երթուղու երկարությունը ամենաշատը $c[p_1]$ է,
- p_0, p_1, p_2 երթուղու երկարությունը ամենաշատը $c[p_2]$ է,
- ...
- $p_0, p_1, p_2, \dots, p_t$ երթուղու երկարությունը ամենաշատը $c[p_t]$ է:

Այս տարի տոնակատարություններ համար ընտրվել են երկու քաղաքներ X -ը և Y -ը: Փակման ժամանակների ընտրության համար **հարմարավետության աստիճանը** սահմանվում է, որպես հետևյալ երկու թվերի գումար.

- X քաղաքից հասանելի քաղաքների քանակ:
- Y քաղաքից հասանելի քաղաքների քանակ:

Նկատել, որ եթե քաղաքը հասանելի է X , և Y քաղաքներից, դա հաշվվում է *երկու* անգամ:

Ձեր խնդիրն է հաշվել հնարավոր ամենամեծ հարմարավետության աստիճանը: Խնդիրը մուլտիթեստային է:

Իրականացման մանրամասներ

Դուք պետք է ծրագրավորեք հետևյալ ֆունկցիան.

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

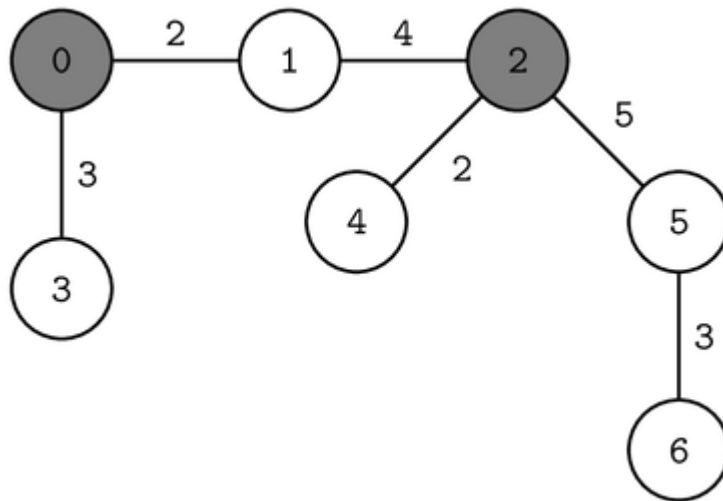
- N . քաղաքների քանակը:
- X, Y . քաղաքները, որտեղ տեղի են ունենալու տոնակատարությունները:
- K . փակման ժամանակների գումարի վերևի սահմանը:
- U, V . $N - 1$ երկարության զանգվածներ, որոնք նկարագրում են ճանապարհները:
- W . $N - 1$ երկարության զանգված, որը նկարագրում է ճանապարհների երկարությունները:
- Այս ֆունկցիան պետք է վերադարձնի մեկ թիվ. մաքսիմալ հարմարավետության աստիճանը, որին հնարավոր է հասնել փակման ժամանակների ինչ-որ ընտրությամբ:
- Այս ֆունկցիան կարող է կանչվել ամեն թեստում **մի քանի անգամ**:

Օրինակ

Դիտարկենք `max_score` ֆունկցիաի հետևյալ կանչը.

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

Այս կանչը համապատասխանում է հետևյալ գրաֆին.



Ենթադրենք փակման ժամանակները ընտրվել են հետևյալ ձևով.

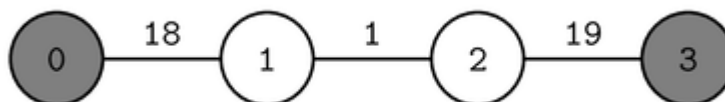
Քաղաք	0	1	2	3	4	5	6
Փակման ժամանակ	0	4	0	3	2	0	0

Նկատեք, որ փակման ժամանակների գումարը 9 է, որը փոքր է կամ հավասար $K = 10$ -ից: 0, 1, և 3 քաղաքները հասանելի են X ($X = 0$) քաղաքից, իսկ 1, 2, և 4 քաղաքները հասանելի են Y ($Y = 2$) քաղաքից: Այդ պատճառով հարմարավետության աստիճանը ստացվում է $3 + 3 = 6$: Գոյություն չունի փակման ժամանակների ավելի լավ ընտրություն, այդ պատճառով ֆունկցիան պետք է վերադարձնի 6:

Դիտարկենք նաև հետևյալ կանչը.

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

Այս կանչը համապատասխանում է հետևյալ գրաֆին.



Ենթադրենք փակման ժամանակները ընտրվել են հետևյալ ձևով.

Քաղաք	0	1	2	3
Փակման ժամանակ	0	1	19	0

0 քաղաքը հասանելի է X ($X = 0$) քաղաքից, իսկ 2 և 3 քաղաքները հասանելի են Y ($Y = 3$) քաղաքից: Այսինքն, հարմարավետության աստիճանը ստացվում է $1 + 2 = 3$: Գոյություն չունի փակման ժամանակների ավելի լավ ընտրություն, այդ պատճառով ֆունկցիան պետք է վերադարձնի 3:

Սահմանափակումներ

- $2 \leq N \leq 200\,000$
- $0 \leq X < Y < N$
- $0 \leq K \leq 10^{18}$
- $0 \leq U[j] < V[j] < N$ ($0 \leq j \leq N - 2$)
- $1 \leq W[j] \leq 10^6$ ($0 \leq j \leq N - 2$)
- Հնարավոր է հասնել ցանկացած քաղաքից ցանկացած այլ քաղաք օգտագործելով ճանապարհները:
- $S_N \leq 200\,000$, որտեղ S_N -ը max_score ֆունկցիայի կանչերի N -երի գումարն է:

Ենթախնդիրներ

Գրաֆը կանվանենք **գիծ** եթե i -րդ ճանապարհը միացնում է i և $i + 1$ քաղաքները ($0 \leq i \leq N - 2$).

1. (8 միավոր) X և Y քաղաքների միջև երթուղու երկարությունը չի գերազանցում $2K$ -ն:
2. (9 միավոր) $S_N \leq 50$, գրաֆը գիծ է:
3. (12 միավոր) $S_N \leq 500$, գրաֆը գիծ է:
4. (14 միավոր) $S_N \leq 3\,000$, գրաֆը գիծ է:
5. (9 միավոր) $S_N \leq 20$
6. (11 միավոր) $S_N \leq 100$
7. (10 միավոր) $S_N \leq 500$
8. (10 միավոր) $S_N \leq 3\,000$
9. (17 միավոր) Լրացուցիչ սահմանափակումներ չկան:

Գրեյդերի նմուշ

Թող C -ն լինի max_score ֆունկցիայի կանչերի քանակը: Գրեյդերի նմուշը մուտքային տվյալները կարդում է հետևյալ ձևաչափով՝

- տող 1՝ C

որին հետևում է C դեպքերի նկարագրությունները:

Գրեյդերի նմուշը կարդում է դեպքի նկարագրությունը հետևյալ ձևաչափով՝

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

Ամեն դեպքի համար գրեյդերի նմուշը տպում է մեկ տող հետևյալ ձևաչափով՝

- տող 1՝ `max_score` ի վերադարձրած թրվը