B. Хос хотууд

Time limit	2 s
Memory limit	512 MB

Өгүүлбэр

Индонезид 0 - ээс $N-1$ хүртлэх тоонуудаар дугаарлагдсан N тооны хотууд байдаг. Мөн 0 - ээс $M-1$ хүртлэх тоонуудаар дугаарлагдсан M тооны хоёр чиглэлтэй замууд байгаа. Зам бүр хоёр ялгаатай хотуудыг холбоно. i-р зам нь $U[i]$-р болон $V[i]$-р хотуудыг холбох ба машинаар явахад $W[i]$ нэгж түлш зарцуулдаг. Аль ч хоёр хотуудын хооронд замууд ашиглан аялж болдог байхаар хотуудыг холбосон байна.

Q өдрийн турш өдөр бүр нэг хос хотын хооронд улстөрийн харилцаа үүсгэхийг хүсдэг. Тухайлбал, j-р өдөр $X[j]$-р хот $Y[j]$-р хоттой улстөрийн харилцаа үүсгэхийг хүснэ. Ингэхийн тулд $X[j]$-р хот $Y[j]$-р хот руу төлөөлөгчөө машинаар илгээх хэрэгтэй. Мөн, $Y[j]$-р хот ч $X[j]$-р хот руу төлөөлөгчөө машинаар илгээх хэрэгтэй юм.

Бөглөрлөөс зайлсхийхийн тулд хоёр машин хэзээ ч уулзахгүй байх ёстой. Тухайлбал, хоёр машин хоёулаа нэг зэрэг нэг хотод байж болохгүй юм. Мөн уг хоёр машин нэгэн зэрэг нэг замаар эсрэг чиглэлд явж болохгүй. Бас машин тодорхой нэг замаар явж байгаа бол уг замаа дуусгаж төгсгөлд нь байгаа хотод очих ёстой (өөрөөр хэлбэл машинууд замын голоос 180 градус эргэж буцаж явж болохгүй гэсэн үг). Харин машин нэг хот эсвэл зам дээгүүр нэгээс олон удаа явж болно. Мөн машинууд нь аль ч хотод хэзээ ч хүлээлт хийж болно.

Их түлшний багтаамжтай машин үнэтэй байдаг тул хоёр хот машинуудынхаа хамгийн их багтаамж нь хамгийн бага байхаар замуудаа сонгохыг хүсч байгаа. Хот бүрт хязгааргүй түлш бүхий станцууд байгаа тул ямар нэг машинд шаардлагатай түлшний багтаамж нь түүний аялсан замуудын түлшний хэрэглээнүүдийн **хамгийн их** нь байна.

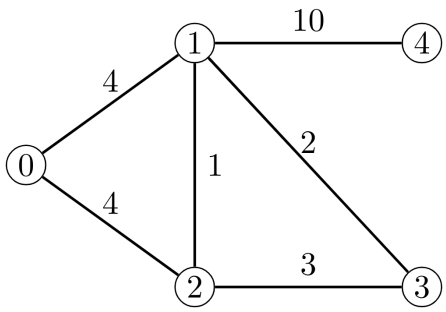
Даалгавар

Та `init` ба `getMinimumFuelCapacity` функцуудыг хэрэгжүүлэх ёстой.

- `init(N, M, U, V, W)` - Уг функцийг шалгагч `getMinimumFuelCapacity`-ийн дуудалтуудаас өмнө яг нэг удаа дуудна.
  - N: Хотуудын тоог илэрхийлэх нэг бүхэл тоо.
  - M: Замуудын тоог илэрхийлэх нэг бүхэл тоо.
  - U: Замуудын эхний төгсгөлүүдийг илэрхийлэх M тооны бүхэл тоо агуулах массив.
  - V: Замуудын эцсийн төгсгөлүүдийг илэрхийлэх M тооны бүхэл тоо агуулах массив.
  - W: Замуудын түлшний хэрэглээг илэрхийлэх M тооны бүхэл тоо агуулах массив.
- `getMinimumFuelCapacity(X, Y)` - Энэ функцийг шалгагч яг Q удаа дуудна.
  - X: Эхний хотыг илэрхийлэх нэг бүхэл тоо.
  - Y: Хоёр дахь хотыг илэрхийлэх нэг бүхэл тоо.
  - Уг функц нь бодлогын өгүүлбэрт бичсэн дүрмийн дагуу X-р хотын төлөөлөгч Y-р хотод очдог, Y-р хотын төлөөлөгч X-р хотод очдог байхаар хоёр машины түлшний багтаамжийн хамгийн их утгын хамгийн бага утгыг илэрхийлэх нэг бүхэл тоог буцаах ба харин ингэх боломжгүй бол -1 тоог буцаана.

Жишээ

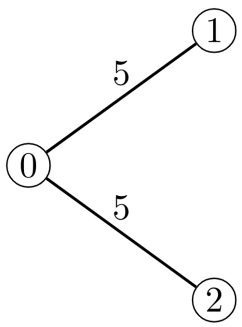
Эхний жишээнд $N = 5, M = 6, U = [0, 0, 1, 1, 1, 2], V = [1, 2, 2, 3, 4, 3], W = [4, 4, 1, 2, 10, 3], Q = 3, X = [1, 2, 0], Y = [2, 4, 1]$ байна. Уг жишээг доорх зургаар дүрсэлж болно:



Шалгагч эхлээд `init(5, 6, [0, 0, 1, 1, 1, 2], [1, 2, 2, 3, 4, 3], [4, 4, 1, 2, 10, 3])` дуудалтыг хийнэ. Үүний дараа шалгагч доорх дуудалтуудыг хийнэ:

- `getMinimumFuelCapacity(1, 2)`. Нэгдүгээрт, эхний хотоос гарсан машин гуравдугаар хот руу очиж чадна. Хоёрдугаарт, хоёрдугаар хотоос гарсан машин нэгдүгээр хот руу очиж чадах ба гуравдугаар хотоос гарсан машин хоёрдугаар хот руу очиж чадна. Иймд хоёр машины хувьд хамгийн их түлшний багтаамж нь 3 нэгж түлш байна (гуравдугаар хотоос хоёрдугаар хот руу очиход шаардагдана). Үүнээс бага түлшний багтаамж шаардагдах замууд байхгүй тул уг функц нь 3утгыг буцаана.
- `getMinimumFuelCapacity(2, 4)`. Дөрөвдүгээр хот руу ирж байгаа эсвэл тэндээс явж байгаа ямар ч машин 10 нэгж түлшийг шаардах тул уг функц нь 10 утгыг буцаана.
- `getMinimumFuelCapacity(0, 1)`. Уг функц 4 утгыг буцаана.

Хоёр дахь жишээнд $N = 3, M = 2, U = [0, 0], V = [1, 2], W = [5, 5], Q = 1, X = [1], Y = [2]$ байна. Уг жишээг доорх зургаар дүрсэлж болно:



Шалгагч эхлээд `init(3, 2, [0, 0], [1, 2], [5, 5])` дуудалтыг хийнэ. Үүний дараа шалгагч доорх дуудалтыг хийнэ:

- `getMinimumFuelCapacity(1, 2)`. Эхний хотоос хоёр дахь хот руу явсан машин нөгөө машинтайгаа уулзалгүйгээр очих боломжгүй тул уг функц нь -1 утгыг буцаана.

Хязгаарлалт

- $2 \le N \le 100\,000$.
- $N - 1 \le M \le 200\,000$.
- $0 \le U[i] < V[i] < N$.
- Хос хот бүрийн хувьд дээд тал нь нэг зам байна.
- Ямар ч хоёр хотын хооронд замуудыг ашиглан зорчиж болно.

- $1 \leq W[i] \leq 10^9$.
- $1 \leq Q \leq 200\,000$.
- $0 \leq X[j] < Y[j] < N$.

Дэд бодлого 1 (6 оноо)

- Хот бүр дээд тал нь хоёр замын төгсгөл байж болно.

Дэд бодлого 2 (7 оноо)

- $M = N - 1$.
- $U[i] = 0$.

Дэд бодлого 3 (17 оноо)

- $Q \leq 5$.
- $N \leq 1\,000$.
- $M \leq 2\,000$.

Дэд бодлого 4 (20 оноо)

- $Q \leq 5$.

Дэд бодлого 5 (23 оноо)

- $M = N - 1$.

Дэд бодлого 6 (27 оноо)

- Нэмэлт хязгаарлалт байхгүй.

Жишээ Шалгагч

Жишээ шалгагч нь оролтыг доорх хэлбэрээр уншина:

```
N M
U[0] V[0] W[0]
U[1] V[1] W[1]
.
.
.
U[M-1] V[M-1] W[M-1]
Q
X[0] Y[0]
X[1] Y[1]
.
.
.
X[Q-1] Y[Q-1]
```

`getMinimumFuelCapacity` - ийн дуудалт бүрийн хувьд жишээ шалгагч нь уг функцийн буцаасан утгыг хэвлэнэ.