

International Olympiad in Informatics 2013

July 2013 6-13

Brisbane, Australia



رؤیا
Persian — ۱۰۰

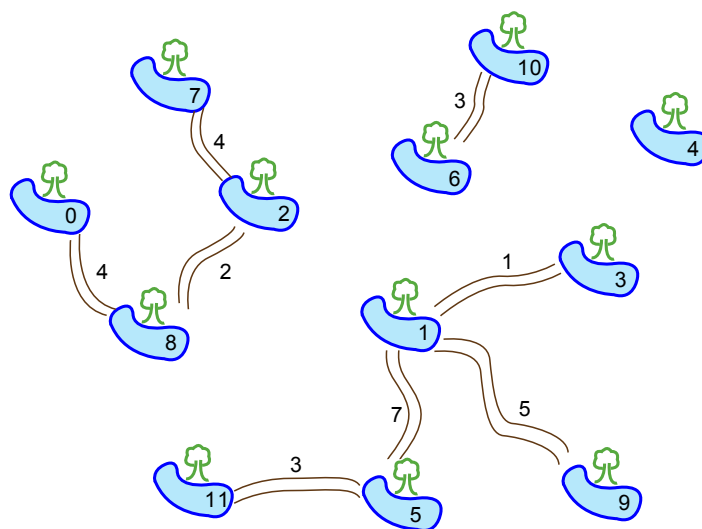
این داستان در سال‌های خیلی دور رخ می‌دهد؛ زمانی که دنیا تازه به وجود آمده بود و کسی IOI را در رؤیا هم نمی‌دید.

یک مار در سرزمینی زندگی می‌کند که N برکه دارد. برکه‌ها با اعداد 0 تا $N-1$ شماره‌گذاری شده‌اند. M جاده‌ی دوطرفه این برکه‌ها را به هم متصل می‌کند و مار می‌تواند روی این جاده‌ها حرکت کند. هر زوج برکه با حداکثر یک دنباله از جاده‌ها (به صورت مستقیم یا غیر مستقیم) به هم متصل شده‌اند، هرچند برخی زوج برکه‌ها می‌توانند اصلاً به هم متصل نباشند (بنابراین $M \leq N-1$). عبور از هر جاده تعداد مشخصی روز برای مار طول می‌کشد. این تعداد می‌تواند برای هر جاده متفاوت باشد.

کانگورو، دوست مار، می‌خواهد $N-M-1$ جاده‌ی جدید بسازد طوری که مار بتواند بین هر زوج برکه حرکت کند. کانگورو می‌تواند بین هر دو برکه‌ی دل‌خواه جاده بسازد و پیمایش هر جاده‌ی جدیدی که کانگورو می‌سازد L روز برای مار طول می‌کشد.

علاوه بر این، کانگورو می‌خواهد که حرکت مار را تا حد ممکن سریع کند. بنابراین او جاده‌های جدید را طوری می‌سازد که بیش‌ترین زمان حرکت بین هر دو برکه کم‌ترین مقدار ممکن را داشته باشد. شما باید به کانگورو کمک کنید که جاده‌ها را به طریق گفته‌شده بسازد و بعد از ساخته‌شدن جاده‌ها، بیش‌ترین زمان حرکت بین برکه‌ها را بیابد.

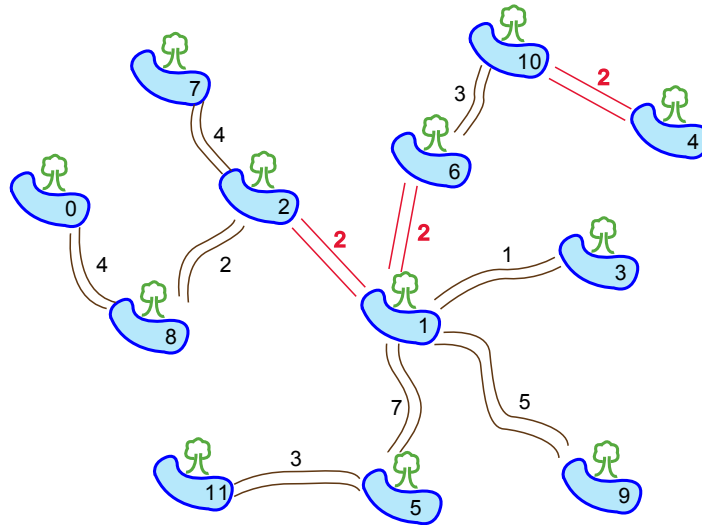
مثال‌ها



در شکل بالا $N = 12$ برکه و $M = 8$ جاده وجود دارد. فرض کنید عبور از هر جاده‌ی جدید $L = 2$ روز برای مار طول می‌کشد. حال کانگورو می‌تواند سه جاده‌ی جدید به صورت زیر بسازد:

▪ بین برکه‌های ۱ و ۲

- بین برکه‌های ۱ و ۶
- بین برکه‌های ۴ و ۱۰



شکل بالا مجموعه‌ی نهایی جاده‌ها را نشان می‌دهد. بیش‌ترین زمان حرکت در این شکل ۱۸ روز است که بین برکه‌های ۰ و ۱۱ رخ می‌دهد. این کم‌ترین جواب ممکن است: مستقل از این که کانگورو چه جاده‌هایی را بسازد، دو برکه وجود خواهند داشت که حرکت بین آن‌ها برای مار حداقل ۱۸ روز طول می‌کشد.

پیاده‌سازی

شما باید تابع `travelTime()` را به شکل زیر در یک فایل پیاده‌سازی و ارسال کنید.

تابع شما: `travelTime()`

C/C++

```
int travelTime(int N, int M, int L,
               int A[], int B[], int T[]);
```

Pascal

```
function travelTime(N, M, L : LongInt;
                   var A, B, T : array of LongInt) : LongInt;
```

توضیحات

این تابع باید بیش‌ترین زمان حرکت (به روز) بین هر زوج برکه را محاسبه کند، با این فرض که کانگورو $N - M - 1$ جاده را طوری اضافه کرده است که همه‌ی برکه‌ها به هم متصل شوند و بیش‌ترین زمان حرکت بین برکه‌ها کم‌ترین مقدار ممکن را داشته باشد.

پارامترها

- `N`: تعداد برکه‌ها.
- `M`: تعداد جاده‌های اولیه که از قبل وجود دارند.

- L : تعداد روزهایی که طول می‌کشد تا مار یک جاده‌ی جدید را ببیماید.
- A ، B و T : آرایه‌هایی به طول M که نقاط پایانی و زمان پیمایش جاده‌های موجود را مشخص می‌کند، طوری که i امین جاده برکه‌های $A[i-1]$ و $B[i-1]$ را به هم متصل می‌کند و زمان پیمایش آن در هر یک از دو جهت $T[i-1]$ روز است.
- خروجی: بیش‌ترین زمان حرکت بین هر زوج برکه، همان طور که در بالا توضیح داده شد.

اجرای نمونه

اجرای زیر مثال بالا را توصیف می‌کند:

Parameter	Value
N	12
M	8
L	2
A	[0, 8, 2, 5, 5, 1, 1, 10]
B	[8, 2, 7, 11, 1, 3, 9, 6]
T	[4, 2, 4, 3, 7, 1, 5, 3]
Returns	18

محدودیت‌ها

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۶۴ مگابایت
- $1 \leq N \leq 100,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10,000$
- $1 \leq L \leq 10,000$

زیرمسئله‌ها

زیرمسئله	امتیاز	محدودیت‌های ورودی اضافی
۱	۱۴	$M = N - 2$ و به هر برکه دقیقاً یا یک یا دو جاده‌ی اولیه منتهی می‌شود. به عبارت دیگر، دو مجموعه از برکه‌های همبند وجود دارند و در هر مجموعه جاده‌ها یک مسیر بدون انشعاب را تشکیل می‌دهند.
۲	۱۰	$M = N - 2$ و $N \leq 100$
۳	۲۳	$M = N - 2$
۴	۱۸	به هر برکه حداکثر یک جاده‌ی اولیه منتهی می‌شود.
۵	۱۲	$N \leq 3,000$
۶	۲۳	(بدون محدودیت اضافی)

آزمایش

مصحح نمونه روی کامپیوتر شما ورودی را از فایل `dreaming.in` می‌خواند که محتوای آن به شکل زیر است:

■ خط ۱: `N M L`

■ خطوط ۲ تا $M + 1$: `A[i] B[i] T[i]`

به طور نمونه، مثال بالا باید به شکل زیر در ورودی داده شود:

```
12 8 2
0 8 4
8 2 2
2 7 4
5 11 3
5 1 7
1 3 1
1 9 5
10 6 3
```

نکات زبان

C/C++ عبارت `#include "dreaming.h"` را باید به برنامه اضافه کنید.

Pascal شما باید `unit Dreaming` را تعریف کنید. تمام آرایه‌ها از 0 (و نه 1) شروع می‌شوند.

برای دیدن مثال‌ها به راه‌حل‌های نمونه (برروی کامپیوتر خود) مراجعه کنید.