



## International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

# dreaming

Czech — 1.0

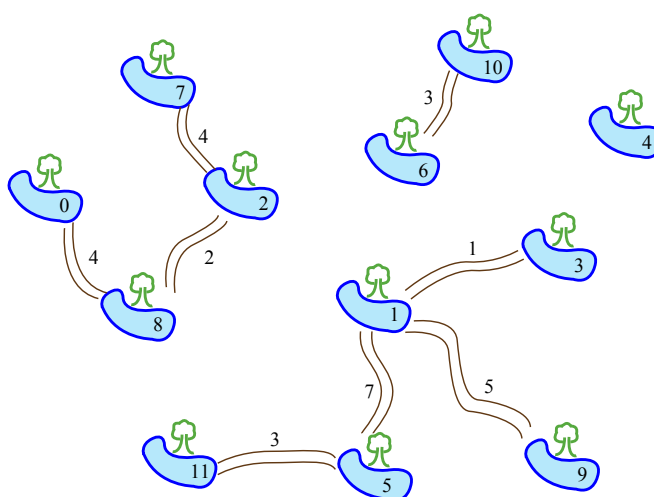
Před dávnými a dávnými časy, když svět byl ještě v záruce a sen o IOI ještě nebyl zrozen...

V krajině s  $N$  billabongy (studnami), očíslovanými  $0, \dots, N-1$ , žije Had. Mezi studnami je  $M$  obousměrných *silnic*, spojujících dvojice studen. Had může cestovat pouze po silnicích. Mezi každými dvěma studnami vede nejvýše jedna cesta, tedy posloupnost jedné či několika navazujících silnic. Všimněte si, že z toho plyne  $M \leq N-1$ . Mezi některými studnami ale nemusí vést žádná cesta. Proplazení přes silnici zabere Hadovi vždy stejný počet dní, tento počet se ale může lišit pro různé silnice.

Hadův přítel Klokan chce postavit  $N - M - 1$  nových silnic tak, aby Had mohl cestovat mezi libovolnými dvěma studnami. Klokan může postavit silnici mezi libovolnými dvěma studnami a proplazení se přes každou takovou novou silnici Hadovi zabere  $L$  dní.

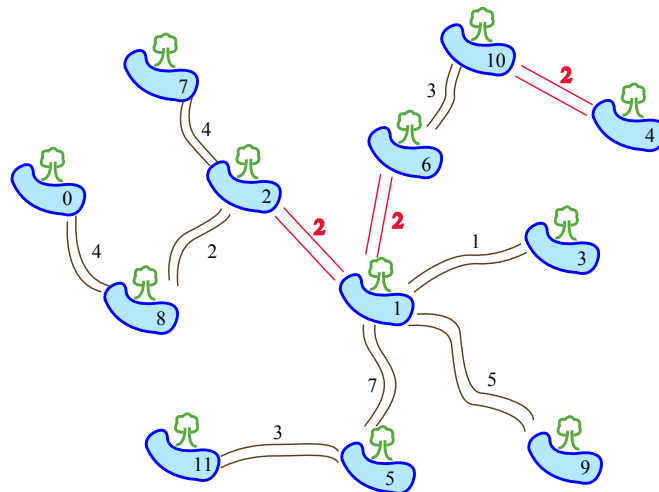
Klokan chce silnice postavit tak, aby Had mohl cestovat co nejrychleji, tedy tak, aby nejdelší ze všech cest mezi dvojicemi studen zabrala co nejméně času. Řekněme, že Klokan postaví cesty optimálním způsobem. Určete čas, který Had bude potřebovat k proplazení takové nejdelší cesty.

### Příklad



V předchozím obrázku je  $N = 12$  studen a  $M = 8$  silnic. Necht'  $L = 2$ , takže přeplazení každé z nových silnic hadovi zabere 2 dny. Klokan může postavit následující tři silnice:

- mezi studnami 1 a 2;
- mezi studnami 1 a 6;
- mezi studnami 4 a 10.



Předchozí obrázek ukazuje situaci po postavení těchto silnic. Nejdelší cesta je mezi studnami 0 a 11 a zabere Hadovi 18 dní. Toto je optimální - at' už Klokan postaví nové silnice jakkoliv, budou existovat dvě studny takové, že cesta mezi nimi zabere Hadovi alespoň 18 dní.

---

## Implementace

Odevzdejte soubor implementující funkci `travelTime()` s následujícími parametry:

**Vaše funkce:** `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;
```

### Popis

Tato funkce určí délku nejdelší cesty (ve dnech, které Had stráví jejím proplazením) mezi dvojicí studen, za předpokladu, že Klokan postavil  $N - M - 1$  silnic tak, aby taková nejdelší cesta byla co nejkratší.

### Parametry

- `N`: počet studen.
- `M`: počet již existujících silnic.
- `L`: počet dní, které Had potřebuje k přeplazení každé Klokanem postavené nové cesty.
- `A`, `B` a `T`: jednorozměrná pole velikosti `M` určující studny spojené existujícími silnicemi a délky těchto silnic v počtu dní, které Had stráví jejich přeplazením. Silnice číslo `i` spojuje studny `A[i-1]` a `B[i-1]` a Hadovi její přeplazení zabere `T[i-1]` dní (kterýmkoliv směrem).
- *Návratová hodnota*: Počet dní, které Had stráví plazením se mezi nejvzdálenějšími studnami v optimálním řešení, jak bylo popsáno dříve.

---

## Příklad

Následující tabulka ukazuje parametry ve výše popsaném příkladu.

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

## Omezení

- Časový limit: 1 sekunda
- Paměťový limit: 64 MiB
- $1 \leq N \leq 100\,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10\,000$
- $1 \leq L \leq 10\,000$

## Podúlohy

Podúloha	Bodů	Dodatečné předpoklady
1	14	$M = N - 2$ a každá studna sousedí s jednou či dvěma již existujícími silnicemi. Jinak řečeno, vstup obsahuje dvě skupiny propojených studen a obě tyto skupiny jsou nerozvětřující se cesty.
2	10	$M = N - 2$ a $N \leq 100$
3	23	$M = N - 2$
4	18	Každá ze studen sousedí s nejvýše jednou již existující silnicí.
5	12	$N \leq 3\,000$
6	23	(žádné)

---

## Testování

Vzorový vyhodnocovač, který najdete v adresáři úlohy, načítá vstup ze souboru `dreaming.in`. Tento soubor musí mít následující formát:

- řádek 1: `N M L`
- řádky 2, ..., `M + 1`: `A[i] B[i] T[i]`

Předchozímu příkladu odpovídá následující vstupní soubor:

```
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
```

---

## Poznámky k jednotlivým programovacím jazykům

**C/C++** You must `#include "dreaming.h"`.

**Pascal** You must define the `unit Dreaming`. All arrays are numbered beginning at `0` (not `1`).

See the solution templates on your machine for examples.