International Olympiad in Informatics
2013
6-13 July 2013
Brisbane, Australia

dreaming

Japanese — 1.0

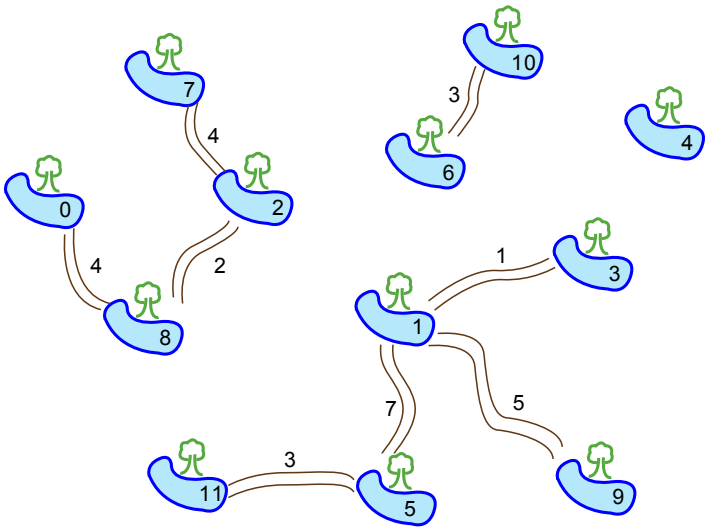
この物語は，まだ世界ができたばかりで，IOI など夢にも見られていなかった遥か昔の出来事である．

ヘビ (Serpent) は $0, \cdots, N-1$ の番号が付いた N 個の**水たまり** (billabongs) からなる土地に住んでいる．M 本の**道** (trails) があり，各道は 2 個の水たまりを結んでいる．ヘビは道に沿って双方向に移動できる．どの 2 個の水たまりについても，それらを (直接的または間接的に) 繋ぐ道の列は高々 1 通りである．全く繋がっていないような 2 個の水たまりがあるかもしれない．したがって，$M \leq N-1$ である．各道は，ヘビが移動するためにある日数がかかる．この日数は道によって異なるかもしれない．

ヘビの友達であるカンガルー (Kangaroo) は，$N-M-1$ 本の新しい道を作り，ヘビがどの水たまりからどの水たまりへも移動できるようにしようと考えた．カンガルーはどの 2 個の水たまりの間にも道を作ることができる．また，カンガルーが作るどの道も，ヘビが移動するために L 日がかかる．

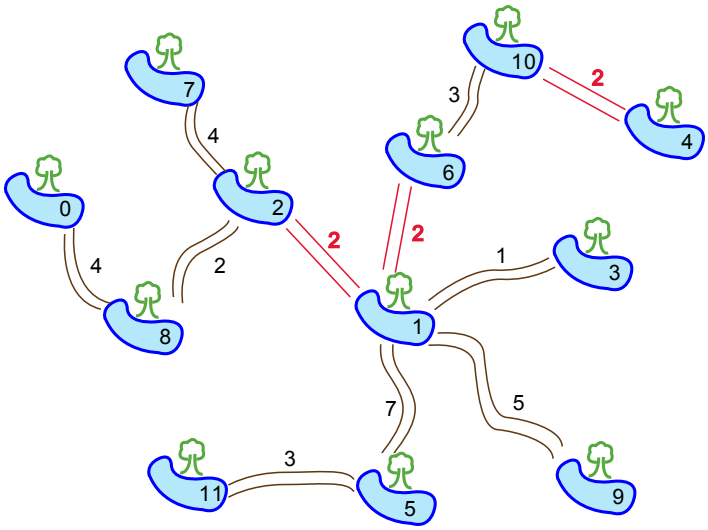
さらに，カンガルーはヘビがなるべく早く移動できるようにしたい．カンガルーは，2 個の水たまりの間の移動日数の最大値が最小になるように新しい道を作る．カンガルーとヘビの手伝いとして，カンガルーがこの方法で新しい道を作ったときの，2 個の水たまりの間の移動日数の最大値を求めよ．

例 (Examples)



上図では，$N = 12$ 個の水たまりと $M = 8$ 本の道がある．$L = 2$，すなわち，どの新しい道もヘビが移動するために 2 日かかるとする．カンガルーは，次のように 3 本の新しい道を作ることができる：

- 水たまり 1 と 2 の間,
- 水たまり 1 と 6 の間,
- 水たまり 4 と 10 の間.



上図はこのときの最終的な道の様子を示している．移動日数の最大値は水たまり 0 と 11 の間であり，18 日かかる．これは最小の可能な結果である — カンガルーがどのように道を作っても，ヘビが移動するために 18 日以上かかるような 2 個の水たまりが存在する．

実装 (Implementation)

あなたは，次のような関数 `travelTime()` を実装した 1 つのファイルを提出しなければならない．

あなたの関数 (Your Function): `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;
```

説明 (Description)

この関数は，カンガルーがすべての水たまりが繋がっているように，かつ 2 個の水たまりの間の移動日数の最大値が最小になるように $N - M - 1$ 本の道を作ったときの，その移動日数を計算しなければならない．

引数と戻り値 (Parameters)

- `N`：水たまりの個数.
- `M`：既に存在する道の本数.
- `L`：ヘビが新しい道を移動するためにかかる日数.
- `A`, `B`, `T`：既に存在する道の端点と移動日数を表す長さ `M` の配列. `i` 番目の道は，水たまり `A[i-1]` と `B[i-1]` を結び，どちらの方向へ移動するのにも `T[i-1]` 日がかかる.
- **Returns** (戻り値): 上で説明された通りの，2 個の水たまりの間の移動日数の最大値.

やりとりの例 (Sample Session)

次のやりとりは，上に述べた例を表す.

引数	値
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]`
戻り値	`18`

制限 (Constraints)

- 時間制限：1 秒
- メモリ制限：64 MiB
- `1 ≤ N ≤ 100,000`
- `0 ≤ M ≤ N - 1`
- `0 ≤ A[i], B[i] ≤ N - 1`
- `1 ≤ T[i] ≤ 10,000`
- `1 ≤ L ≤ 10,000`

小課題 (Subtasks)

小課題	得点	入力に関する追加の制約
1	14	$M = N - 2$ であり，各水たまりからは既に存在する道が 1 本または 2 本出ている．言い換えると，道で繋がっている水たまりの集合が 2 個あり，それぞれの集合では道は分岐しない 1 本のパスをなす．
2	10	$M = N - 2$ かつ $N \leq 100$
3	23	$M = N - 2$
4	18	各水たまりからは既に存在する道が高々 1 本しか出ていない．
5	12	$N \leq 3{,}000$
6	23	(なし)

試行 (Experimentation)

与えられる採点プログラムのサンプルは，ファイル `dreaming.in` から入力を読み込む．このファイルは次のフォーマットで書かれていなければならない：

- 1 行目：`N M L`
- 2, …, $M + 1$ 行目：`A[i] B[i] T[i]`

例えば，上に述べた例は次のフォーマットで与えられなければならない：

```
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
```

言語に関する注意 (Language Notes)

C/C++ `#include "dreaming.h"` を行うこと．

Pascal `unit Dreaming` を定義すること．すべての配列は `0` から始まる (`1` からではない)．

実装例については，あなたの計算機上に与えられている解法テンプレートを参照すること．