

International Olympiad in Informatics

2013

July 2013 6-13

Brisbane, Australia



dreaming

Hebrew — 1.0

הסיפור קרה מזמן, בעולם צעיר לפני שחלמו על ה - IOI.

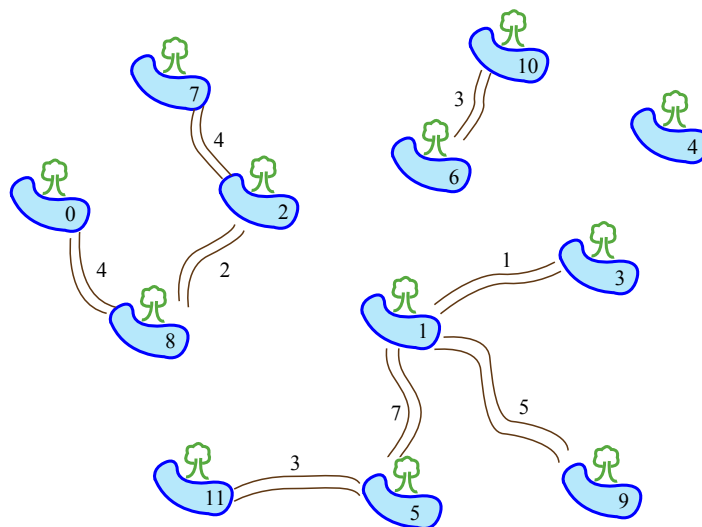
הנחש חי בארץ שבה יש N אגמים הממוספרים $0, \dots, N-1$. יש M שבילים דו-כיווניים שכל אחד מהם מחבר זוג אגמים. הנחש הולך לאורך השבילים. כל זוג אגמים מחובר (באופן ישיר או עקיף) על-ידי רצף אחד של שבילים לכל היותר, אבל ייתכנו גם זוגות אגמים שאינם מחוברים בכלל (לכן מתקיים $M \leq N-1$). עבור כל שביל, נתון מספר הימים שלוקח לעבור אותו. המספר הזה עשוי להיות שונה לשבילים שונים.

(Serpent = נחש) (Billabong = אגם) (Trail = שביל)

הקנגורו רוצה לבנות $N-M-1$ שבילים חדשים, כך שהנחש יוכל להגיע מכל אגם לכל אגם. הקנגורו יכול לבנות שביל בין כל זוג אגמים, וכל שביל שהקנגורו בונה הוא כזה שלנחש לוקחים L ימים לעבור אותו.

בנוסף, הקנגורו רוצה שהנחש יוכל לטייל כמה שיותר מהר. לכן, הוא רוצה לבנות את השבילים החדשים כך שזמן ההליכה המקסימלי בין זוג אגמים יהיה קצר ככל האפשר. הקנגורו יבנה את השבילים החדשים באופן זה. עליכם לחשב את זמן ההליכה המקסימלי בין שני אגמים, לאחר הבנייה.

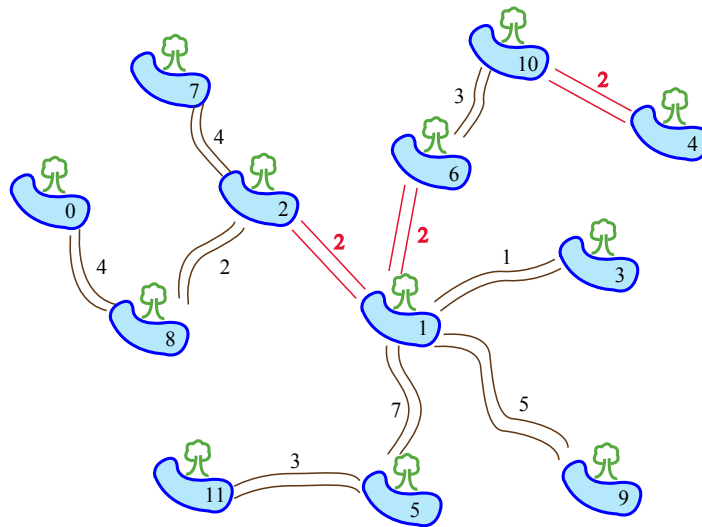
דוגמאות



באיור למעלה יש $N = 12$ אגמים ו- $M = 8$ שבילים. נניח ש- $L = 2$, כלומר, כל שביל חדש שהקנגורו יבנה הוא כזה שנדרשים 2 ימים כדי לעבור אותו. הקנגורו יכול, לדוגמא, לבנות שלושה שבילים חדשים:

▪ בין אגם 1 ואגם 2;

- בין אגם 1 ואגם 6;
- בין אגם 4 ואגם 10;



האיור למעלה מראה את אוסף השבילים הסופי. ההליכה הארוכה ביותר היא של 18 ימים, בין אגם 0 ואגם 11. זו הבנייה האופטימלית, כלומר, בכל דרך שבה הקנגורו יבנה את השבילים, יהיה זוג אגמים שנדרשים לפחות 18 ימים כדי לעבור ביניהם.

Implementation

You should submit a file implementing the function `travelTime()`, as follows:

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;
```

Description

הפונקציה מחשבת את זמן ההליכה המקסימלי (בימים) בין שני אגמים, בהנחה שהקנגורו בנה `N-M-1` שבילים באופן אופטימלי.

Parameters

- `N` : מספר האגמים.
- `M` : מספר השבילים שכבר קיימים בהתחלה.

- L : הזמן בימים שדרוש כדי לעבור כל שביל חדש שהקנגורו בונה.
- A , B and T : מערכים באורך M שמתארים את השבילים כך שהשביל ה- i מחבר בין האגמים $A[i-1]$ ו- $B[i-1]$, ולוקח $T[i-1]$ ימים לעבור אותו.
- הפונקציה מחזירה את מספרים הימים המקסימלי הדרוש כדי לטייל בין שני אגמים, בהנחה שהבנייה אופטימלית.

Sample Session

The following session describes the example above:

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

Constraints

- Time limit: 1 second
- Memory limit: 64 MiB (MiB = 1048576 Bytes)
- $1 \leq N \leq 100,000$
- $0 \leq M \leq N - 1$
- $0 \leq A[i], B[i] \leq N - 1$
- $1 \leq T[i] \leq 10,000$
- $1 \leq L \leq 10,000$

Subtasks

Subtask	Points	Additional Input Constraints
1	14	ובנוסף מתקיים שבהתחלה כל אגם נוגע בדיוק $M = N - 2$ בשביל אחד או בשני שבילים. במילים אחרות, יש שתי קבוצות קשירות של אגמים וכל אחת מהן היא בצורת מסלול בלי התפצלויות.
2	10	$M = N - 2$ and $N \leq 100$
3	23	$M = N - 2$
4	18	בהתחלה, כל אגם נוגע בשביל אחד לכל היותר.
5	12	$N \leq 3,000$
6	23	(None)

הרצות בדיקה

The sample grader on your computer will read input from the file `dreaming.in`, which must be in the following format:

- line 1: `N M L`
- lines 2, ..., $M + 1$: `A[i] B[i] T[i]`

For instance, the example above should be provided in the following format:

```
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
```

Language Notes

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.