

НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

2-4 юни 2023 г.

Група А, 11-12 клас

Задача А1: Скрито дърво

Дадено е скрито дърво с N върха (и $(N - 1)$ ребра). Искаме да намерим структурата му, като можем да питаме въпроси от следния вид:

Въпроса изисква множество от върхове $S = \{v_1, \dots, v_k\}$ и специален връх x , за $2 \leq k \leq N$. Отговора е "да" или "не", в зависимост от това дали има **прост път** съдържащ връх x , като краищата му са върхове от зададеното множество S .

Ще напомним, че прост път в граф е път, който не минава през връх повече от веднъж. Целта на задачата е да открием ребрата в скритото дърво, като питаме горния въпрос колкото се може по-малко пъти.

Детайли по имплементацията

Решението на задачата трябва да в във функция `solve`, имплементирана от вас и имаща следния прототип:

```
void solve(int n);
```

Тя ще бъде извикана 1 път за всеки тест и ще получи като аргумент броя върхове в дървото N . Върховете са номерирани с числата $1, \dots, N$. За задаване на въпроси, е предоставена функцията `query`, имаща следния прототип:

```
bool query(std::vector<int> S, int x);
```

Аргументите на тази функция са съответно S и x . Тя връща `true` или `false` в зависимост от това дали отговора на този въпрос е "да" или "не". Функцията трябва да бъде извиквана с валидни върхове (номерирани от 1 до N), като върховете в S не трябва да се повтарят, и трябва да има поне 2 върха в това множеството.

След като откриете скритото дърво, функцията `solve` трябва да извика функцията `answer` точно $(N - 1)$ пъти. Всяко извикване на `answer` съответства на някое ребро от дървото, като прототипа е даден под:

```
void answer(int u, int v);
```

Редът на ребрата няма значение (реда на извикване на `answer`), както и реда на краищата (u и v за всяко извикване). Програмата ви също така трябва да включва хедър файла `treeq.h`, който съдържа описаните прототипи:

```
#include "treeq.h"
```

НАЦИОНАЛЕН ЛЕТЕН ТУРНИР ПО ИНФОРМАТИКА

2-4 юни 2023 г.

Група А, 11-12 клас

Ограничения

- $2 \leq N \leq 2000$

Подзадачи и оценяване

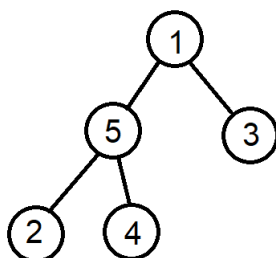
Подзадача	Точки	Допълнителни ограничения
1	10	$N = 50$
2	50	$N = 250$
3	40	$N = 2000$

Всяка задача съдържа няколко теста, като ако вашата програма не намира дървото, или пита невалиден въпрос, ще получите 0 точки. Частта от точките, които ще получите на дадена подзадача зависи от максималния брой заявки Q , които правите на подтест:

$$\min \left(1.0, \sqrt{\frac{2N \lceil \log_2(N) \rceil}{Q}} \right)$$

Пример

Скритото дърво можем да видим на фигурата долу:



За това дърво, примерна интеракция със системата може да бъде:

Състезател	Жури
	<code>solve(5)</code>
<code>query({1, 5}, 2)</code>	<code>false</code>
<code>query({2, 4}, 5)</code>	<code>true</code>
<code>query({2, 4, 5}, 5)</code>	<code>true</code>
<code>query({2, 4, 5}, 3)</code>	<code>false</code>
<code>answer(1, 5)</code>	
<code>answer(2, 5)</code>	
<code>answer(5, 4)</code>	
<code>answer(1, 3)</code>	