





ЕСЕНЕН ТУРНИР ПО ИНФОРМАТИКА

София, 24 – 26 ноември 2023 г.

Група А, 11 – 12 клас

 : 3,0 сек.
 : 256 MB

Задача А3. ЛИСица

ЛИСицата намери къщата на малкия Праси. Вместо да я срути, както направи вълкът, тя реши да открадне най-ценното притежание на Праси - а именно най-голямото число, което той притежава. Праси има малък масив, който съдържа пермутация на числата от 0 до $n - 1$. За да намери къде е $n - 1$, ЛИСицата може да казва на Праси да разбърка числата в своя масив по даден начин и след това да намери дължината на най-дългата нарастваща подредица¹ в него. За нейно щастие, Праси не е подредено прасе и затова е гарантирано, че първоначалният масив е произволно разбъркан.

Помогнете на ЛИСицата да намери позицията на $n - 1$ в началния масив.

¹ *Нарастваща подредица в масив е поредица от индекси $i_1 < i_2 < \dots < i_k$, такива че $a_{i_1} < a_{i_2} < \dots < a_{i_k}$. Най-дългата нарастваща подредица е тази с най-голямо k .*

Задача

Напишете програма `lis`, съдържаща функция `find_maximum`, която ще се компилира с програмата на журито и ще комуникира с нея, задавайки въпроси от гореописания вид. В края на изпълнението си тя трябва да е открила позицията на максимума в масива преди разбъркванията.

Детайли по имплементацията

Функцията `int find_maximum(int n)`, която трябва да напишете, ще бъде извикана T на брой пъти от програмата на журито и като аргумент ще получи цялото число n . За комуникация с програмата на журито Ви се предоставят следната функция:

```
int get_lis_length(const std::vector<int>& p);
```

При всяко извикване на функцията `get_lis_length`, тя ще върне дължината на най-дългата нарастваща подредица на $a_{p_0}, a_{p_1}, \dots, a_{p_{n-1}}$. Векторът трябва да е пермутация на $0, 1, \dots, n - 1$. Сложността на изпълнение на функцията е $O(n \log(n))$.

Вашата програма `lis.cpp` трябва да имплементира функцията `find_maximum(int n)`. Тя може да съдържа и друг код, и функции, необходими за работата Ви, но не трябва да съдържа главната функция `main`. Също така, не трябва да четете от стандартния вход или да отпечатвате на стандартния изход. Програмата Ви трябва да включва хедър файла `lis.h` чрез указание към предпроцесора:

```
#include "lis.h"
```

Ограничения

- $8 \leq n \leq 200$
- $0 \leq a_i \leq n - 1$
- a е произволно генерирана пермутация, тоест всички възможни входове за дадено n са еднакво вероятни.
- $T = 5$ на брой подтестове



ЕСЕНЕН ТУРНИР ПО ИНФОРМАТИКА

София, 24 – 26 ноември 2023 г.

Група А, 11 – 12 клас

: 3,0 сек.
 : 256 MB

Подзадачи

Подзадача	Точки	N
1	5	= 8
2	35	= 20
3	60	= 200

Точките за дадена подзадача се получават само ако се преминат успешно всички тестове, предвидени за нея.

Оценяване

- Нека q е най-големият броят на повиквания на `get_lis_length`, които вашата програма прави за един подтест.
- За първата подзадача, ако отговорите Ви за всички подтестове са верни, получавате 100% от точките.
- За втората и третата подзадача резултатът Ви се изчислява по следния начин:

$$score = \begin{cases} 1.0, & \text{ако } q \leq target \\ 0.9 \times \left(\frac{\log(target)}{\log(q)}\right)^{1.5}, & \text{иначе} \end{cases}$$

където

$$target = \begin{cases} 2 \cdot n^2, & \text{за втората подзадача} \\ 2 \cdot n, & \text{за третата подзадача} \end{cases}$$

Примерна комуникация

Функция на участника	Програма на журито
	<code>find_maximum(3)</code>
<code>get_lis_length({0, 1, 2})</code>	2
<code>get_lis_length({0, 2, 1})</code>	2
<code>get_lis_length({2, 0, 1})</code>	1
<code>return(2)</code>	

Пояснение: Началната редица е $\{1, 0, 2\}$.

Локално тестване

Предоставен Ви е файлът `Lgrader.cpp`, който може да компилирате заедно с вашата програма, за да я тествате. При стартиране програмата ще чете от стандартния вход стойността на T , след това следват T на брой теста. За всеки тест се въвежда n и след това началната редица a_0, \dots, a_{n-1} . Ако решението е вярно, ще получите изход `Correct` и броя завки, които е направила програмата Ви. Може да модифицирате предоставения файл, както искате.