

ТРЕНИРОВЪЧНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

София, 3 юли 2023 г.

Група G, 9 – 12 клас

Задача GT31. КЛЕТКИ

Вие сте световноизвестен биолог, който е успял да изолира много важни за науката клетки. Те са K на брой и се намират на различни позиции с цели координати от 1 до N . Проблемът е, че не знаете на кои позиции се намират клетките, понеже те са микроскопични. За щастие, можете да използвате вещество, което се привлича от най-близките клетки. Почти можете да сложите веществото на цяло число от 1 до N и ще получите позициите на едната или двете най-близки клетки до него (ако има две равноотдалечени от веществото).

Времето е важно за вашето изследване и затова искате да намерите клетките с възможно най-малко операции.

Задача

Напишете програма `cells`, която да намира позициите на всички клетки. Тя трябва да съдържа функцията `find_cells`, която ще се компилира с програма на журито.

Детайли по реализацията

Функцията `find_cells` трябва да е в следния формат:

```
std::vector < int > find_cells (int N, int K);
```

Тя се вика веднъж от програмата на журито с два параметъра – съответно броят на позициите и броят на клетките. Функцията трябва да върне вектор, съдържащ позициите на намерените клетки, без значение от тяхната подредба.

Функцията, чрез която можете да намерите най-близки клетки, има следния формат:

```
std::vector < int > get_closest_cells (int pos);
```

Параметърът `pos` е позицията, на която искате да сложите веществото. Тя връща вектор с позициите на най-близките клетки до тази позиция, тоест всички клетки, за които $|cell_i - pos|$ е минимално. Когато има две най-близки клетки, те се връщат в нарастващ ред. Ако не се спазва $1 \leq pos \leq N$, то ще получите `Wrong Answer` за текущия тест. Сложността на функцията е $O(K)$. Вашата програма може да вика функцията неограничен брой пъти.

Вашата програма `cells.cpp` трябва да имплементира функцията `find_cells`. 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 `cells.h` чрез указание към препроцесора:

```
#include "cells.h"
```

Ограничения

- $100 \leq N \leq 10^9$
- $K = 20$
- $1 \leq cell_i \leq N, cell_i \neq cell_j$ за $i \neq j$

ТРЕНИРОВЪЧНО СЪСТЕЗАНИЕ НА РАЗШИРЕНИЯ НАЦИОНАЛЕН ОТБОР

София, 3 юли 2023 г.

Група G, 9 – 12 клас

Подзадачи

Подзадача	Точки	N	K
1	10	$= 100$	$= 20$
2	90	$\leq 10^9$	$= 20$

Оценяване

За първата подзадача няма ограничение на броя заявки и всяко решение, което върне правилен отговор, в рамките на ограниченията по време и памет, получава максималния брой точки за нея.

За втората подзадача, нека означим броя извиквания на `get_closest_cells` с q :

- Ако $q \leq 40$, то програмата Ви получава 100% от точките за подзадачата.
- Иначе, получавате точки по формулата:

$$\max\left(1 - \frac{\sqrt{\ln \frac{q}{40}}}{4}, 0\right) \cdot 90$$

Локално тестване

За локално тестване са предоставени файловете `cells.h` и `Lgrader.cpp`. Сложете Вашия файл `cells.cpp` и двата предоставени файла в една папка. Като компилирате само `Lgrader.cpp`, ще получите програма, с която ще проверите верността на функциите Ви. Програмата ще изисква от стандартния вход следната последователност от данни:

- на първия ред две цели положителни числа – броят позиции N и броят клетки K .
- на втория ред K цели положителни числа – позициите на клетките.

Ако не спазвате протокола за комуникация или намерите грешни клетки, ще получите подходящо съобщение. Иначе, при успешно изпълнение на програмата, ще получите съобщение “*Correct with q queries used.*”, където q е броят заявки, които програмата Ви е използвала.

Примерна комуникация

№	Действия на вашата програма	Действия и отговори на журито
1.		<code>find_cells(10, 2)</code>
2.	<code>get_closest_cells(5)</code>	<code>return {7}</code>
3.	<code>get_closest_cells(8)</code>	<code>return {7, 9}</code>
4.	<code>return {7, 9}</code>	