

Задача 1. ПОДРЕДИЦИ 2.0

1 sec. 256 MB

Колко от Вас са решавали Increasing от контроли за разширения младши отбор 2022? Нека направим една ретроспекция.

Боби се чудил как да подготви възможно най-добрите тестове за задачата, като Сашка му дала идея - да може ефикасно да генерира пермутации с малка дължина, които да имат точно определен брой нарастващи подредици. Едно ненаредено подмножество позиции $S = \{i_1, i_2, \dots, i_{|S|}\}$ на пермутация p_0, p_1, \dots, p_{N-1} на числата от 0 до $N - 1$ е нарастваща подредица, когато:

- По дефиниция за ненаредено множество, $i_x < i_{x+1}, 1 \leq x < |S|$.
- $0 \leq i_x \leq N - 1$.
- $p_{i_x} < p_{i_{x+1}}, 1 \leq x < |S|$

Празното множество **се брои** за нарастваща подредица. Сашка ще поиска от Боби Q на брой пъти да генерира пермутация, като i -тия път ще иска тя да има точно k_i нарастващи подредици. Напишете програма `increasing2`, която да генерира пермутации с малък брой елементи N .

Детайли по имплементацията

Задачата е дадена в интерактивен вариант, в който Вие трябва да имплементирате функция, която генерира пермутация по дадено K .

Функцията `std::vector<int> construct_permutation(long long K)`, която трябва да напишете, ще бъде извикана Q на брой пъти от програмата на журито и като аргумент ще получи цялото число K , равно на k_i за текущото питане на Сашка.

Функцията трябва да върне вектор, представляващ пермутацията, където на i -та позиция да съдържа елемента на i -та позиция.

Ограничения

- $1 \leq Q \leq 100$
- $1 \leq k_i \leq 10^{18} (1 \leq i \leq Q)$

Подзадачи и оценяване

1. (0 точки) Примерният тест
2. (10 точки) $1 \leq k_i \leq 90$
3. (90 точки) Без допълнителни ограничения

За първа и втора подзадача се получават пълен брой точки при правилно конструирани пермутации с до 90 елемента за всеки тест и 0 за всякакъв вид други свинщини.

За трета подзадача има частично оценяване на максималното N_{max} , което сте ползвали за конструиране на пермутация:

- За $N_{max} \leq 90$ се получават 90 точки
- За $90 < N_{max} \leq 120$ се получават $90 - \frac{N_{max}-90}{3}$ точки
- За $120 < N_{max} \leq 5000$ се получават $80 - \frac{N_{max}-120}{65}$ точки
- За $5000 < N_{max}$ се получават 0 точки

Локално тестване

Предоставен Ви е файлът `Lgrader.cpp`, който може да компилирате заедно с вашата програма, за да я тествате.

- На първия ред: Q
- На $1 + i$ -тия ред: k_i .

На изход ще получите върнатите пермутации от програмата Ви.

Примерен вход и изход на Lgrader

Вход	Изход	Обяснение на примера
2	2	(1) Нарастващи подредици са \emptyset , $\{0\}$, $\{1\}$. (2) Всяка подредица е нарастваща.
3	1 0	
8	3 0 1 2	