Мутації

Твердження: кількість ітерацій моделювання процесу мутації, до того моменту як геном буде мати вид [1, 0, 0, 0….0] (**0**), можна обмежити зверху як log(log(N)) (**1**)

Моделювання однієї ітерації може бути просто реалізовано за лінійну складність відносно N за допомогою використання допоміжного масиву. Після кожної ітерації треба перевірити, чи перетворився геном на (**0**). Якщо це так, то далі моделювання не має сенсу — геном буде залишатися незмінним. В результаті програма виконає мінімальне з O(N\*K) та O(N \* (**1**)) операцій.

Можете ознайомитись з псевдокодом, що моделює один етап мутації геному з N елементів, що зберігається в массиві А і записує результат у массив В.

b[] = [0, 0, …, 0]

for i = 1 to N do

 b[a[i]] = b[a[i]] + 1

Доведення (1):

Розглянемо дві ітерації моделювання мутації (А => В => С). Нехай z(X) — кількість ненульових елементів в геномі Х. Очевидно, що кількість z(B) — кількість різних ненульових елементів в А.

Доведемо, що z(C) <= (z(A)^0.5) \* K, де К — певна константа. За змістом z(C) — це кількість різних частот серед частот геному А. Щоб отримати P різних частот потрібно хоча б P\*(P + 1) / 2 елементів(1+2+3..+..Р). З цього слідує, що W елементів можуть породити не більше O(W^0.5) елементів через два кроки мутації. Отже, після двох мутацій N-елементного геному кількість його ненульових елементів стає не більша за O(N^0.5).

Нехай f(N) — кількість мутацій, які потрібно здійснити, щоб N-елементний геном перетворився на (**0**). Маємо f(1) = O(1), а f(N) = f(N^0.5) + 1. При заміні числа на його квадратний корінь, логарифм цього числа стає вдічі меншим. Кількість ітерацій ділення на два, поки число не стане рівне О(1) дорівнює логарифму цього числа, а оскільки в ролі цього “числа” виступає log(N), то маємо (**1**).

Слід відмітити, що коли z(Q) = 1, де Q —певний геном, то він в будь якому випадку перетвориться на (**0**), тому за одну ітерацію він перетвориться на [0, 0, …, 1, 0, 0, …, 0], а він вже, в свою чергу, перетвориться на (**1**).

#include <iostream>

using namespace std;

long n,k,t,s;

long a[200000],b[200000];

int main()

{

cin>>n>>k;

for(int i=1;i<=n;i++)

{

 cin>>t;

 b[t]++;

 }

for(int j=1;j<k;j++)

{

 for(int i=1;i<=n;i++){

 a[b[i]]++;

 }

s=0;

 for(int i=1;i<=n;i++){b[i]=a[i];a[i]=0;s=s+b[i];}

if(s==1 && b[1]==1)break;

}

for(int i=1;i<=n;i++)

 cout<<b[i]<<" ";

 cout << endl;

 return 0;

}