

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА Д-Р ВАДИЗЪБОВ

Вероятно първото хрумване на всеки, който се захваща със задачата, е да подреди данните за пациентите в хронологичен ред на пристигането им. Нека и ние го направим ;-) И сега остава да се проследи какво се случва в кабинета при пристигане на пореден пациент:

- Ако пред кабинета има опашка от чакащи, дошлият пациент се нарежда в края на опашката, което води до увеличаване на крайния момент за обслужване на опашката с времето, необходимо за обслужване на новодошлия;
- Ако пред кабинета няма чакащи и в момента докторът не обслужва пациент, новодошлият веднага сяда на стола и крайният момент за обслужване на опашката (чакащ за обслужване е само новодошлия) се формира от момента на идването и времето за обслужване на новодошлия.

Т.е. при обработване на данните в хронологичен ред трябва да следим само величината „краен момент за обслужване на опашката чакащи“ в съответствие с по-горните две правила! Няма нужда да правим пълна имитация на опашката чакащи с класически решения за АТД „опашка“! Накрая получаваме отговора на задачата като разлика на крайния момент време за обслужване на опашката и момента на идването на последния пациент.

В допълнение да споменем, че заради малкия диапазон на моментите на пристигане (най-много може да има 721 пациента) можем да приложим линеен метод на сортиране на входните данни: още при прочитане на данните за пациент ги поставяме в елемент на масив с индекс, съвпадащ с момента на идването.

Сорс-кодът на програмата

```
#include <iostream>
using namespace std;

char A[721];
short lastmoment,i,j,n;

int main () {
    for (;cin>>i>>j;n++) A[i]=j;
    for (i=0; n; i++) if (A[i]) {
        if (i>=lastmoment) lastmoment=i;
        lastmoment+=A[i];
        n--;
    }
    cout<<lastmoment-i+1<<endl;
    return 0;
}
```

Автор: Евгений Василев