

# КУРС „ДИЗАЙН И АНАЛИЗ НА АЛГОРИТМИ“

## летен семестър 2008

### СЪСТЕЗАНИЕ 1 – АНАЛИЗИ

#### ЗАДАЧА 1 – СПИРАЛА

В тази задача трябва да се забележи начина, по който се образува поредния квадрат. Той винаги има страна колкото сумата от страните на предните два образувани квадрата. Това означава, че страните на тези квадрати са числата на Фибоначи. Числата от редицата на Фибоначи можем да дефинираме по следния начин като функция с един цял параметър:

$$F(0) = 0, F(1) = 1,$$
$$F(n) = F(n-1) + F(n-2), \text{ за всяко } n > 1,$$

Следователно задачата се свежда до това да се пресметнат числата на Фибоначи до някое поредно и да се пресметне общото лице. Трябва да се внимава с това, че за по-големите стойности на  $N$  отговора става прекалено голямо число и не се побира в 32-битово число например. За целта трябва да се използват по-големи типове като *long long* в *C/C++* и *long* в *Java*. Често при задачи с толкова малко възможни входни данни си струва да се изпечатат всички отговори и да се провери дали изглеждат добре. Дори да не е лесно да се провери дали числата са правилни, може да се забележи, че някои са отрицателни, когато се използва по-малък тип.

Друго важно наблюдение е че ако не успеем да се сетим каква е зависимостта между числата от отговора е полезно да напишем на ръка първите няколко и да видим дали не ни напомнят на нещо познато.

#### ЗАДАЧА 2 – АМА ЧЕ СОРТИРАНЕ!

Едно възможно решение на задачата е да се опитаме да наредим числата едно след друго като започнем от най-голямото и завършим с най-малкото. Това означава първо да поставим на мястото му най-голямото число, след това второто най-голямо и така нататък докато не наредим всичко числа. За да се постигне това може да мислим отзад напред. За да застане най-голямото число на последна позиция, където му е мястото, преди да застане там то трябва да е било на първа позиция. Това е така, защото единствената операция, която имаме е обръщане на числата от първото до някое. Следователно ако в крайна сметка най-голямото число се е озовало на последна позиция, значи по някое време е било на първа и сме обърнали всички числа. Ако преди да отиде то на първа позиция е било на позиция  $X$ , то е достатъчно да обърнем числата от 1 до  $X$ . Ето така излиза, че с два хода можем да закараме най-голямото число на последна позиция, където му е мястото. След като сме го поставили там можем да се погрижим за числата, които стоят наляво от него. То вече е на мястото си и не се грижим за него повече. Правим абсолютно същото за второто най-

голямо число, за да отиде то на предпоследна позиция и т.н. за всички числа. Един пример как ще наредим една редица от числа:

*Позициите номерираме от 1 до 5*

- 1) 3, 8, 2, 5, 6 – правим ход 1..2, защото 8 е най-голямото число и искаме да наредим него
- 2) 8, 3, 2, 5, 6 – правим ход 1..5, за да закараме 8 на последното място
- 3) 6, 5, 2, 3, 8 – сега ще подреждаме 6, то вече е на първа позиция, следователно правим ход 1..4, за да отиде на предпоследна позиция
- 4) 3, 2, 5, 6, 8 – следващото число е 5, то вече е там където трябва да бъде
- 5) 3, 2, 5, 6, 8 – правим ход 1..2, за поставим 3 там където му е мястото
- 6) 2, 3, 5, 6, 8 – с това сме готови

### **ЗАДАЧА 3 – ЦИКЛИ**

В задачата се иска да се провери дали има цикъл в граф. Графа описва селото, като улиците са негови ребра, а кръстовищата - върхове. Тази задача спада към темата за търсене в дълбочина. Идеята беше да се реши като от всеки връх в графа се пусне обхождане в дълбочина и ако дадения връх е достижим сам от себе си да се отговори, че има цикъл в графа. Това означава в най-лошия случай да пуснем  $N$  пъти обхождането. Това води до сложност  $O(N * (N + E))$ , където  $E$  е броят ребра в графа. Максималният брой ребра в графа е от порядъка на  $N^2$ . Следователно при ограничението от 100 за броя върхове тази сложност е допустима.

Трябва да се отбележи, че е важно да се внимава при обхождането да не се правят скокове от вида: връх  $A$  към връх  $B$ , после връх  $B$  към връх  $A$ . Това не се брои за цикъл, защото в условието се изисква да се върви само по различни улици.

### **ЗАДАЧА 4 – ПОЗНАНСТВА**

В задачата се иска хората и познанствата да се представят като граф, в който да се преброи от каква степен е всеки връх. Т.е. колко познати има всеки човек. След това върховете да се сортират по степента си в намаляващ ред. За целта може да се използва както сортиране с квадратна сложност, така и по-бързо сортиране със сложност  $O(N \cdot \log N)$ . Това е така, защото броя върхове в графа е не повече от 200.

### **ЗАДАЧА 5 – ИНТЕРВАЛИ!**

Растяща редица и търсене в нея трябва веднага да ни подсетят за двоично търсене. За всеки интервал с двоично търсене трябва да се намери къде стои едното и къде другото число от него. След това лесно може да се види колко числа има между тях. Понеже редицата е строго растяща няма повтарящи се числа. Ако имаше трябва да се внимава за точното определяне на това кои числа са в зададения интервал.