

**Домашна работа № 2 по “Дизайн и анализ на алгоритми”**  
**за специалност “Компютърни науки”, 2. курс, 1. поток — СУ, ФМИ,**  
**летен семестър на 2017 / 2018 уч. г.**

**СЪСТАВЯНЕ НА АЛГОРИТМИ**

<i>Задача</i>	1	2	3, а	3, б	3, в	<i>Общо</i>
<i>получен брой точки</i>						
<i>максимален брой точки</i>	20	20	20	20	20	100

**Задача 1.** Функцията  $f(n)$  е дефинирана за цели неотрицателни числа:

$$f(0) = 0, \quad f(1) = 1, \quad f(2n) = f(n), \quad f(2n + 1) = f(n) + f(n + 1).$$

Съставете алгоритъм за изчисляване на  $f(n)$  със сложност по време  $O(\log n)$  и сложност по памет  $O(1)$ . Опишете алгоритъма като програма на езика C. Анализирайте сложността на алгоритъма по време и памет.

**Упътване:** Опитайте се да изчислите  $f$  от конкретна стойност, напр.  $f(99)$ , чрез развиване на уравнението. Видът на получаваните изрази ще ви подскаже идеята за решението.

**Задача 2.** В един град всички улици са еднопосочни. В града има  $n$  кръстовища, номерирани с целите числа от 1 до  $n$ . Някои двойки кръстовища  $p$  и  $q$  са свързани с еднопосочна отсечка от улица, така че по тази отсечка от улица няма други кръстовища. За всеки две кръстовища  $p$  и  $q$  има най-много една еднопосочна отсечка от улица — или в посока от  $p$  към  $q$ , или от  $q$  към  $p$ . При спазване на правилата за движение в града невинаги е възможно да отидем с кола от кръстовище  $a$  до кръстовище  $b$ . С колко най-малко нарушения можем да се придвижим от  $a$  до  $b$ ? Всяко навлизане в посока, обратна на разрешената, в еднопосочна отсечка от улица се брой за едно нарушение.

Ако  $n$  е броят на кръстовищата, а  $m$  е броят на еднопосочните отсечки от улици, които ги свързват, то алгоритъм с времева сложност  $O(m + n)$  носи 20 точки, алгоритъм с времева сложност  $O((m + n) \log(m + n))$  се оценява с 10 точки, а по-бавен алгоритъм не носи точки.

**Задача 3.** Когато искаме да съединим голяма купчина листове с телбод, може да се наложи да използваме няколко телчета. Например, ако всяко телче захваща максимум 10 листа, а ние искаме да съединим 25 листа, ще трябва да използваме три телчета: с едното съединяваме листовите от № 1 до № 10, с второто — листовите от № 10 до № 19, с третото — от № 19 до № 25. Има и други начини, например с първото телче можем да защитим листовите от № 1 до № 10, с второто — от № 8 до № 17, с третото — от № 16 до № 25. Тоест не е задължително всички телчета да защитят еднакъв брой листове. Ако обаче с първото телче защитим листовите с номера от № 1 до № 10, с второто телче — от № 12 до № 21, а с третото — от № 19 до № 25, то не всички листове ще бъдат съединени; по-точно, ще има три “папки”: едната папка ще съдържа листовите от № 1 до № 10, втората ще се състои от единствен лист (№ 11), а третата папка ще бъде съставена от листовите с номера от № 12 до № 25.

Тези наблюдения дават повод да разгледаме следната алгоритмична задача: Имаме два масива  $a_1, a_2, \dots, a_n$  и  $b_1, b_2, \dots, b_n$  от цели положителни числа, като  $a_i < b_i$  за всяко  $i = 1, 2, 3, \dots, n$ . Нека  $L = \max \{b_1, b_2, \dots, b_n\}$ . Тълкуваме входните данни като  $n$  телчета, които защитят общо  $L$  листа, номерирани с целите числа от 1 до  $L$ . При това,  $i$ -тото телче защита листовите с номера от  $a_i$  до  $b_i$  включително. Условието  $a_i < b_i$  гарантира, че всяко телче защита поне два листа. Няма горна граница за броя на листовите, които едно телче може да защита. Съставете алгоритми, които да пресмятат броя на получените папки (множества от пряко или косвено захванати листове) и да отговарят на следните изисквания към сложността:

- а) Предложете алгоритъм с максимална времева сложност  $\Theta(n)$ , когато числото  $L$  е от порядъка на  $n$ .
- б) Предложете алгоритъм, който независимо от стойността на  $L$  има времева сложност  $O(n \log n)$  в най-лошия случай.

*Упътване:* Използвайте алгоритъма от подусловие “а”.

- в) Съществува ли алгоритъм, основан на сравнения, който независимо от стойността на  $L$  работи във време  $O(n)$  в най-лошия случай?

Опишете алгоритмите с думи или на псевдокод. Илюстрирайте ги с примери. Анализирайте техните максимални времеви сложности.