Изпит по "Дизайн и анализ на алгоритми" (поправка — СУ, ФМИ, юни 2016 г.)

Име: ________________________________ ФН:_____ Спец.:____ Курс:___

Задача	1а	1б	1в	2а	2б	3а	3б	4а	4б	5	Общо
получени точки											
максимум точки	5	5	10	10	10	20	10	20	20	20	130

Забележка: За отлична оценка са достатъчни 100 точки.

Задача 1. Да се решат рекурентните уравнения с точност до порядък:

а) $T(n) = T(n-1) + \sqrt[7]{n^{25}}$; б) $T(n) = 625\, T\left(\dfrac{n}{5}\right) + 6n^3$; в) $T(n) = \sum\limits_{k=1}^{n-1} k \,.\, T(k)$.

Задача 2. В небостъргач има няколко асансьора, всеки от които спира само на някои етажи. На един етаж може да спират няколко асансьора. Всеки асансьор може да вози и нагоре, и надолу. Предложете бърз алгоритъм, намиращ маршрут от един етаж до друг, ако искаме да стигнем:

а) за най-малко време (всички асансьори се движат с еднаква скорост);

б) с най-малък брой прекачвания.

Задача 3. Пътник трябва да стигне от град C_0 до град C_n, като мине през всеки от градовете C_1 , C_2 , ... , C_{n-1} непременно в този ред. За всеки участък от маршрута пътникът може да избира между две транспортни компании. Превозът от C_{k-1} до C_k $(k = 1, 2, \ldots, n)$ струва A_k лева с първата компания и X_k лева с втората. Двете компании имат по-ниски цени за продължение на пътуването: съответно B_k лева с първата компания и Y_k лева с втората; $B_k < A_k$, $Y_k < X_k$ $(k = 2, 3, \ldots, n)$. "Продължение" значи, че в участъка от C_{k-2} до C_{k-1} и в участъка от C_{k-1} до C_k пътникът ползва услугите на един и същи превозвач.

а) Съставете алгоритъм `OptimalTransport` $\big(A[1\ldots n],\ B[2\ldots n],\ X[1\ldots n],\ Y[2\ldots n]\big)$, който за време $\Theta(n)$ намира най-ниската цена за пътуване от C_0 до C_n .

б) Разширете алгоритъма така, че да казва с кой превозвач да бъде изминат всеки участък, та общата цена да бъде възможно най-ниска.

Упътване: Използвайте динамично програмиране с числова таблица `dyn`$[1\ldots n][1\ldots 2]$, където `dyn`$[k][i]$ е най-ниската възможна цена за пътуване от C_0 до C_k , ако последният участък от пътя (т.е. от C_{k-1} до C_k) бъде пропътуван с i-тата компания.

Задача 4. Дадени са три масива от цели положителни числа $L[1\ldots n]$, $W[1\ldots n]$ и $H[1\ldots n]$, които съдържат размерите на n играчки с формата на правоъгълен паралелепипед; тоест k-тата играчка има размери $L[k] \times W[k] \times H[k]$. Известно е, че играчките могат да бъдат вложени една в друга като редица от матрьошки, но не непременно в същия ред, в който са дадени техните размери.

а) Предложете алгоритъм с времева сложност $\mathrm{O}(n \log n)$, който подрежда играчките в реда на тяхното влагане (първа е най-вътрешната играчка, последна — най-външната).

б) Докажете, че всеки алгоритъм, който подрежда играчките в реда на тяхното влагане, изисква време $\Omega(n \log n)$.

Задача 5. Да се докаже, че е NP-трудна следната алгоритмична задача:
"За даден граф G и дадено цяло положително число k да се разпознае дали G притежава покриващо дърво, всички върхове на което имат степени, ненадвишаващи k."

РЕШЕНИЯ

Задача 1. а) Развиваме уравнението: $T(n) = T(0) + 1^{25/7} + 2^{25/7} + \ldots + n^{25/7} \asymp n^{32/7}$.
б) С мастър-теоремата: $k = \log_5 625 = 4$, $n^{k-\varepsilon} \succ 6n^3$, напр. $\varepsilon = 0{,}01$. Значи, $T(n) = \Theta\big(n^4\big)$.
в) $T(n) = \sum\limits_{k=1}^{n-1} k \,.\, T(k)$. Заместваме n със $n-1$: $T(n-1) = \sum\limits_{k=1}^{n-2} k \,.\, T(k)$. Това уравнение го вадим от оригиналното: $T(n) - T(n-1) = (n-1) \,.\, T(n-1)$, тоест $T(n) = n \,.\, T(n-1)$.
Развиваме полученото уравнение: $T(n) = n(n-1) \,.\, T(n-2) = \ldots = n!\, T(0) = \Theta(n!\,)$.