# Поправителен изпит по Дизайн и анализ на алгоритми, Инф. и КН-2, 05.09.2019 г.

Име: ................................................................ ФN№: ............. Група: .... Спец.: .....

| Зад. | 1 | 2 | 3 | 4 | Общо |
|---|---|---|---|---|---|
| *точки* | | | | | |
| *от макс.* | 25 | 25 | 25 | 25 | 100 |

*Допустимо е да използвате наготово всичко, изучавано на лекции по ДС и ДАА, но ако правите това, напишете ясно за какво става дума.*

---

**Зад. 1** Обяснете какво означава сложност по време на алгоритъм, какво е сложност по време в най-лошия случай, какво е сложност по време в най-добрия случай, и какво е средна сложност по време. (2 точки)

Докажете, че алгоритъмът QUICKSORT има средна сложност по време $\Theta(n \lg n)$. (23 точки)

**Зад. 2** Нека $n \geq 1$ и $I_n = \{1, 2, \ldots, n\}$. Дадено е множество $C \subseteq I_n \times I_n$. Търси се биекция $f : I_n \to I_n$, такава че $\forall (x, y) \in C : f(x) < f(y)$. Предложете ефикасен алгоритъм (10 точки), който по дадено $C$:

- или връща такава биекция $f$, ако такава биекция съществува,
- или връща информация, че такава биекция не съществува, в противен случай.

Допуснете, че такава функция $f$ съществува и вече сте я намерили с алгоритъма от предната подзадача. Нека $X = \{x \in I_n \,|\, \forall y \in I_n : (x, y) \notin C\}$. Първо докажете, че $X \neq \emptyset$ (5 точки). Нека $D$ е рефлексивното и транзитивно затваряне на $C$. Нека, за всяко $i \in I_n$, $Z(i) = \{k \in I_n \,|\, (i, k) \in D\}$. Предложете колкото е възможно по-ефикасен алгоритъм, който по дадено $i \in I_n$ връща $|X \cap Z(i)|$. (10 точки)

**Зад. 3** Колко единици, като функция на $n$, печата следният алгоритъм?

Единици($n \in \mathbb{N}$)
**if** $n = 0$
  print $1$
**else**
  **for** $i \leftarrow 1$ **to** $2^n$
    Единици($n - 1$)

Обосновете добре отговора си.

**Зад. 4** Дадени са $n$ града $c_1$, $c_2$, ..., $c_n$. Някои от тях са свързани със шосета. Шосетата, свързващи град с друг град, също са дадени. ФМИ иска да построи свои филиали в тези градове. Бихме искали във всеки от тях да има по един филиал на ФМИ, но, уви, това е скъпо. ФМИ приема икономичен подход, в който е достатъчно за всеки град $c_i$ в него да има филиал или в негов съседен град $c_j$ да има филиал. $c_j$ е съседен град на $c_i$, ако между тях има шосе.

Допуснете, че цената на построяване на филиал на ФМИ във всеки град е една и съща. Можете ли да преложите ефикасен алгоритъм, който по дадени градове и шосета връща минимална цена, достатъчна за построяване на филиали при даденото изискване? Обосновете добре отговора си.

Скица на решенията. Първата задача е нещо от лекции.
Във втората задача е даден ориентиран граф с върхове $1 \ldots n$ и ориентирани ребра (множеството $C$), като може да има примки. Търсената биекция $f$ е всъщност топологическо сортиране. Такова има тстк графът е даг. На лекции е преподаван алгоритъм с линейна сложност по време, който по даден ориентиран граф определя дали графът е даг и, ако е даг, връща негово топологическо сортиране. Множеството $X$ е множеството от всички сифони на дага. Че то е непразно се доказва елементарно с принципа на Дирихле. $D$ е граф, представляващ релацията на достижимост в $C$. Не е необходимо $D$ да бъде строен, това не се иска; $D$ е необходим само концептуално, за да дефинираме $Z(i)$. За всеки връх $i$, $Z(i)$ са върховете, достижими от $i$. Тогава $X \cap Z(i)$ са сифоните, достижими от $i$. По дадено $i$ с тривиална модификация на DFS можем да намерим в линейно време всички сифони, достижими от $i$.
В третата задача алгоритъмът печата точно $2^{\frac{n(n+1)}{2}}$ единици. Това трябва да се докаже по индукция по $n$. Базата, естествено, е $n = 0$.
В четвъртата задача става дума за алгоритмичната задача Dominating Set. За нея е доказвано на лекции, че е NP-пълна, което означава, че е изключително невероятно да намерим ефикасен алгоритъм. Доказателството за NP-пълнотата на DS е правено на лекции, и го има в сборника с решени задачи. А пропо, никъде не се казва, че графът от градове и шосета е планарен. DS остава NP-пълна и върху планарни графи, но от условието на задачата нямаме право да заключим, че графът е планарен.