

Име: ..... ФН#: ..... Група: .... Спец.: .....

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

**Зад. 1** Даден е ориентиран граф  $G = (V, E)$ , за който е известно, че

- всеки връх има полустепен на входа едно (тоест, има точно едно влизащо ребро);
- съществува връх  $u \in V$ , такъв че за всеки друг връх  $v \in V$  има ориентиран път (тоест, маршрут) от  $u$  до  $v$ .

Предложете ефикасен алгоритъм, който пресмята дали  $G$  е dag или не. “dag” означава ориентиран граф без цикли (контури). Точки ще се дават само на решения с оптимална, в асимптотичния смисъл, сложност. Обосновете накратко коректността и сложността по време на алгоритъма.

Ако  $G$  не е dag, предложете ефикасен алгоритъм, който идентифицира множество  $E' \subseteq E$ , такова че  $|E'|$  е минимално и освен това премахването на ребрата от  $E'$  води до това, че  $G$  става dag. Обосновете накратко коректността и сложността по време на алгоритъма.

**Зад. 2** Вие сте на изпит, на който има  $n$  въпроса  $q_1, q_2, \dots, q_n$ . За всяко  $i$ , въпрос  $q_i$  дава  $p_i$  точки за вярно решение. Вие не виждате всички въпроси едновременно, а те Ви излизат след друг в реда на номерирането. Когато видите даден въпрос, може да отговорите на него или да го прескочите, но, ако го прескочите, губите възможността да му отговорите и така губите точките за него. Въпросите са такива, че Вие можете да отговорите вярно на всеки от тях.

Това обаче не Ви гарантира автоматично  $\sum_{i=1}^n p_i$  точки! Има една уловка: някои от въпросите са дразнещи, и то толкова дразнещи, че ако отговорите на дразнещ въпрос, то задължително ще прескочите няколко следващи въпроса. По-прецизно казано, за всеки въпрос  $q_i$  е дадено някакво неотрицателно  $t_i$ , такова че ако отговорите на  $q_i$ , то задължително ще прескочите въпроси  $q_{i+1}, \dots, q_{i+t_i}$ . Стойностите  $t_i$  са такива, че  $i + t_i$  никога не надхвърля  $n$ .

Предложете колкото е възможно по-ефикасен алгоритъм, който за всеки вход  $(n, p_1, \dots, p_n, t_1, \dots, t_n)$  връща максималния брой точки, който можете да получите. Съвсем накратко обосновете коректността и сложността му по време.

**Зад. 3** Нека  $f_1 : \mathbb{N}^+ \rightarrow \mathbb{N}^+$ ,  $f_2 : \mathbb{N}^+ \rightarrow \mathbb{N}^+$  и  $f_3 : \mathbb{N}^+ \rightarrow \mathbb{N}^+$ . Докажете или опровергайте следното твърдение:

$$f_1(n) + f_2(n) \asymp f_1(n) + f_3(n) \quad \rightarrow \quad f_2(n) \asymp f_3(n) \text{ или } f_2(n) \preceq f_3(n) \text{ или } f_2(n) \succeq f_3(n)$$

**Зад. 4** Вие сте майстор-готвач, който трябва да приготви ядене, разполагайки с  $n$  съставки  $i_1, i_2, \dots, i_n$ . Приготвянето на яденето е изключително просто: избирате някакви съставки, тоест подмножество на  $\{i_1, i_2, \dots, i_n\}$ , слагате ги в тенджерата, загревате ги и готово (тенджерата е достатъчно голяма). Целта Ви е да направите ядене с колкото е възможно повече съставки. Обаче, някои съставки са по двойки несъвместими в някаква степен. По-прецизно казано, за всеки  $j, k$ , такива че  $1 \leq j < k \leq n$ , е дадено реално число  $C_{j,k} \in [0, 1]$ , което изразява несъвместимостта между съставки  $i_j$  и  $i_k$ : ако  $C_{j,k} = 0$ , те са максимално съвместими, ако  $C_{j,k} = 1$ , те са максимално несъвместими, ако  $0 < C_{j,k} < 1$ , то те имат известна несъвместимост, зададена от това число. Качеството на яденето е **сумата** от взаимните несъвместимости между (ненаредените) двойки участващи съставки в него.

Дадена е триъгълната таблица от взаимните несъвместимости между двойките съставки и освен това е дадено число  $p$ . Задачата е да пригответе ядене с поне  $p$  съставки, което да има колкото е възможно по-добро качество (с други думи, колкото е възможно по-малка сума от несъвместимости). Докажете, че тази задача е практически нерешима (intractable) при допускането, че  $P \neq NP$ . За целта покажете, че тази задача—която не е задача за разпознаване—съдържа в себе си подзадача, която е задача за разпознаване и е  $NP$ -пълна.