Контролно ДАА

Изберете две от трите задачи (една е бонус). Всяка задача носи по 10 т. Предложете колкото е възможно по-бързи (в асимптотичен смисъл) и оптимални по памет алгоритми за следните проблеми:

Задача 1. Даден е неориентиран граф. Да се намери цикъл в него (алгоритъмът да отпечатва намерения цикъл или че графът е ацикличен).

За всяка свързана компонента правим обхождане с dfs. Цикъл има ако даден връх u има съсед v, който вече е бил посетен, и v е различен от родителя на връх u. Тази проверка може да се извършва като добавим втори аргумент на функцията: dfs(u, parent).
Сложност: $\Theta(m + n)$.

Задача 2. Върхово покритие на граф се нарича множество от върхове, такова че всяко ребро на графа е инцидентно с поне един връх от множеството. Минимално върхово покритие е върхово покритие с възможно най-малкия брой върхове. Да се намери броя върхове в минимално върхово покритие на дърво.

Нека ch(u) е множеството от децата на връх u,
f(u) - мощността на минимално върхово покритие на поддървото с корен u, като се съдържа в това покритие,
g(u) - мощността на минимално върхово покритие на поддървото с корен u, като НЕ се съдържа в това покритие.
За произволно листо u: $f(u) = 1, g(u) = 0$.
За връх, който не е листо:
$$f(u) = 1 + \sum_{v \in ch(u)} \min(f(v), g(v))$$
$$g(u) = 1 + \sum_{v \in ch(u)} f(v)$$
С динамично програмиране - мемоизация - сложността е $\Theta(n)$.

Задача 3. Даден е речник с думи - низове от латински букви. За константно време може да се проверява дали даден низ е дума от речника. Даден е произволен низ s от латински букви. Да се намери дали низът s може да бъде представен като конкатенация на думи от речника.

Нека $dict(x)$ е булевата функция, която има стойност 1 ако x е дума от речника и 0 в противен случай; n - дължнината на s; $substr(i, j)$ е поднизът на s от позиция i до позиция j включително ($1 \leq i \leq j \leq n$).
С ще означаваме дали substr(1, i) може да се представи като конкатенация на думи от речника.
$f(0) = 1$
$f(i) = \vee_{j=1}^{i} (f(j-1) \wedge dict(substr(j, i)))$

Търсим f(n). Сложност: $\Theta(n^2)$

Контролно ДАА

Задача 1. Да се намери броя на свързаните компоненти в неориентиран граф.

Едно обхождане (в ширина или дълбочина) намира една свързана компонента. Стартираме обхождане за всеки непосетен връх. Броят на компонентите е равен на броя на обхожданията. Сложност: $\Theta(n+m)$

Задача 2. Дадено е множество от n правоъгълника с техните размери. Един правоъгълник може да се постави върху друг ако неговите дължина и ширина са съответно по-малки или равни на дължината и ширината на втория. Не е разрешено въртене на правоъгълниците. Да се намери дължината k на максималната редица от правоъгълници $a_1, a_2, \ldots a_k$, такива че a_i може да се постави върху a_{i+1}. (Даденото множество няма наредба, можем да строим редицата по произволен начин)

Нека размерите на i-тия правоъгълник са (x_i, y_i).
I начин: Сортираме правоъгълниците по x, а при равни x-координати - по y. Задачата се свежда до намиране на най-дълга ненамаляваща подредица (longest non-decreasing subsequence) на получената редица от y-координатите. Сложност: $\Theta(nlgn)$.

II начин: Разглежаме правоъгълниците като върхове на DAG. Има ребро от връх (правоъг.) i до j, ако $x_i \leq x_j$ и $y_i \leq y_j$. Правим топологично сортиране и намираме най-дългия път в графа. Сложност: $\Theta(n^2)$.

Задача 3. Дадени са цените на такси $a_1, a_2, \ldots a_k$ за пропътуване съответно на 1, 2, ... k км ($a_i > 0$). Пътник може да пропътува n км като ги раздели на отсечки с дължини измежду числата 1, 2, ... k. Каква е минималната цена за пътуване n км?

Нека f(i) е минималната цена за пътуване на i км. Търсим f(n).
$f(0) = 0$
$$f(i) = \min_{\substack{1 \leq i \leq n \\ 1 \leq j \leq \min(k,i)}} (f(i-j) + a_j)$$
Сложност: $\Theta(nk)$.