Поправителен изпит по Дизайн и Анализ на Алгоритми, 04.09.2017 г.

Име: .... ФN°: .... Група: .... Спец.: .....

Зад.	1	2	3	4	Общо
точки					
от макс.	10	10	15	5	40

Забележка: към **Задача 3** има 5 точки бонус.

Зад. 1 Даден е масив $A[1, \dots, n]$ от цели числа, където $n \geq 1$. За всяко j и за всяко k, такива че $1 \leq j \leq k \leq n$ казваме, че подмасивът $A[j, \dots, k]$ е *група* тогава и само тогава, когато са изпълнени следните три условия:

$$\forall i \in \{j, j+1, \dots, k-1\} : A[i] \leq A[i+1]$$
$$j = 1 \lor (j > 1 \land A[j-1] > A[j])$$
$$k = n \lor (k < n \land A[k] > A[k+1])$$

Дължината на групата $A[j, \dots, k]$ е $k - j + 1$. Групата $A[j_1, \dots, k_1]$ е *вляво* от групата $A[j_2, \dots, k_2]$ тогава и само тогава, когато $k_1 < j_2$. Очевидно може да има няколко групи с максимална дължина. *Максималната група* е най-лявата група измежду групите с максимална дължина.

Забележка: терминът "група" в тази задача е избран произволно и няма нищо общо други използвания на "група", които може да сте срещали.

4 т. **А)** Предложете итеративен алгоритъм със сложност $O(n)$, който намира максималната група. Входът да бъде масивът $A[1, \dots, n]$, а изходът да бъде наредена двойка индекси (j, k), такива че $A[j, \dots, k]$ е максималната група.

6 т. **Б)** Докажете коректността на предложения алгоритъм чрез инварианта.

Зад. 2 Решете следните рекурентни уравнения:

а) $T(n) = (\log_3 7) T\left(\dfrac{n}{\log_2 5}\right) + n$ б) $S(n) = S(n-2) + \dfrac{1}{n}$

в) $R(n) = 2R(n-1) + \displaystyle\sum_{k=0}^{n} \binom{n}{k}$ г) $Q(n) = Q\left(\dfrac{n}{2}\right) + Q\left(\dfrac{n}{3}\right) + n$

За някои от уравненията няма формален метод за решаване, който да сме изучавали на лекции. За тях е допустимо да ги решите не особено прецизно – просто дайте решение и го обосновете колкото можете по-добре. Ако успеете да ограничите добре решението отдолу и отгоре, в асимптотичния смисъл, ще получите точки, макар и непълни, дори долната и горната граница да са различни. Ако докажете по индукция решението си, ще получите някакъв бонус.

Зад. 3 Трябва да се напечата красиво текст върху страница. Текстът се състои от n думи $w_1, \dots, w_n$, в този ред, с дължини съответно $\ell_1, \dots, \ell_n$. Използва се шрифт тип *monospace*, тоест всички букви, както и шпациите между думите, са с една и съща ширина (думите не съдържат шпации). Всеки ред на страницата съдържа точно m символа, които може да са буквите от азбуката или шпации. Текстът няма нищо друго освен въпросните думи и шпациите между тях; с други думи, няма пунктуация. Всяка от думите е непразна (очевидно) и може да се побере на един ред; с други думи, $\forall i : 1 \leq \ell_i \leq m$. Страницата е достатъчно дълга – можете да допуснете, че има поне n реда и дори всяка дума да е на отделен ред, няма да има проблеми да бъде побран текстът. Празни редове преди текста и в текста няма. Празните редове след текста нямат значение и не ни интересуват.

Текстът трябва да бъде ляво подравнен (това означава, че всеки ред, на който има думи, започва с най-лявата буква на най-лявата дума) и между всеки две думи на един ред трябва да има точно една шпация. Лесно се вижда, че при това ограничение, ако на даден ред са написани думите $w_j, \dots, w_k$, където $1 \leq j \leq k \leq n$, то редът ще завършва с точно $m - \left(\sum_{i=j}^{k} \ell_i\right) - (k-j)$ шпации, които запълват мястото от най-дясната буква на най-дясната дума w_k до края на реда. Това "$(k-j)$" е заради шпациите между думите. Съвкупността от всички шпации в края на редовете, които имат текст, се нарича *бялото поле* (*whitespace* на английски).

Да бъде напечатан тесктът красиво означава да се минимизира бялото поле в следния смисъл. В идеалния случай бяло поле няма и тогава текстът изглежда най-добре, защото всеки ред завършва с буква, а не с шпация. Но в някои случаи (на практика, най-вероятно) бялото поле е неизбежно – това зависи от дължините на думите и ширината на страницата. Ние искаме не просто бялото поле като цяло да е колкото е възможно по-малко, а особено държим да няма редове с много шпации в края. За тази цел минимизираме сумата от **квадратите** на броя на шпациите в края на редовете, които имат текст. Същото нещо, написано формално: ако текстът е разположен върху t реда и ред i, за $1 \leq i \leq t$, завършва с точно $g(i)$ шпации, то искаме сумата $\sum_{i=1}^{t} (g(i))^2$ да е минимална.

Предложете ефикасен алгоритъм, който разполага текста на страницата така, че тази сума да е минимална. Достатъчно е да изчислите само цената на оптималното решение (а не самото разполагане на думите по редове).

Упътване и пояснение. Намерете алгоритъм, изграден по схемата динамично програмиране. Алчен (greedy) алгоритъм за тази задача не бил ефикасен. За пълен брой точки е достатъчно да посочите коректно рекурентно уравнение за цената и да го обосновете добре. Ако посочите и как това уравнение може да имплементира ефикасно, ще получите 5 точки бонус.

Зад. 4 Даден е ориентиран граф $G = (V, E)$. Дефинирана е релацията $R \subseteq V \times V$ така:

$\forall u, v \in V : (u, v) \in R$ тогава и само тогава, когато съществува
ориентиран път (маршрут) от u до v и от v до u

Очевидно, R е релация на еквивалентност. Предложете ефикасен алгоритъм, който получава на входа описание на графа чрез списъци на съседство и връща класовете на еквивалентност на R. Съвсем накратко обосновете коректността на предложения от Вас алгоритъм и кажете каква е сложността му по време.