

ALGORYTMY I STRUKTURY DANYCH

PROGRAMOWANIE DYNAMICZNE

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

1. [1.5 pkt.] Udowodnij, że rekurencyjna procedura obliczająca współczynnik dwumianowy $\binom{n}{k}$ będzie w sumie wywoływana $2\binom{n}{k} - 1$ razy.
2. [1.5 pkt.] Pokaż, jak obliczyć współczynnik dwumianowy $\binom{n}{k}$ używając tablicy pomocniczej $L[0 \dots k]$ oraz $O(1)$ dodatkowych komórek pamięci. Następnie pokaż, jak rozwiązać to zadanie używając jeszcze mniejszej tablicy pomocniczej $L[0 \dots \min(k, n - k)]$ oraz $O(1)$ dodatkowych komórek pamięci.
3. [1 pkt.] Wyznacz najdłuższy wspólny podciąg dla ciągów $AABBCC$ i $BABCBCACA$.
4. [2 pkt.] Pokaż, jak obliczyć długość największego wspólnego podciągu używając tablicy pomocniczej $L[0 \dots 2 \min(n, m)]$ oraz $O(1)$ dodatkowych komórek pamięci. Następnie pokaż, jak rozwiązać to zadanie używając jeszcze mniejszej tablicy pomocniczej $L[0 \dots \min(n, m)]$ oraz $O(1)$ dodatkowych komórek pamięci.
5. [2 pkt.] Pokaż, jak zrekonstruować największy wspólny podciąg tylko na podstawie tablicy kosztów C i ciągów wejściowych X i Y . Twój algorytm powinien działać w czasie $O(m+n)$, gdzie $m = |X|$ i $n = |Y|$.
6. [3 pkt.] Dany jest zbiór n punktów na płaszczyźnie $P = \{p_1, \dots, p_n\}$, które są kolejnymi wierzchołkami wielokąta wypukłego. Przedstaw algorytm, który podzieli ten wielokąt przekątnymi (odcinkami łączącymi niesąsiednie wierzchołki) na $n - 2$ rozłącznych trójkątów, tak aby suma długości wszystkich wewnętrznych odcinków (przekątnych dzielących wielokąt na trójkąty) była minimalna.
7. [3 pkt.] Dany jest zbiór n symboli $S = \{s_1, \dots, s_n\}$ z określonym na nich porządkiem liniowym $\leq$ (dzięki relacji $\leq$ elementy te możemy ze sobą porównywać). Z każdym symbolem jest związane prawdopodobieństwo jego wystąpienia p_i dla $i = 1 \dots n$ (możemy myśleć o tym wystąpieniu jak o prawdopodobieństwie zapytania o dany symbol), gdzie $\sum_{i=1}^n p_i = 1$. Przedstaw algorytm, który wygeneruje drzewo poszukiwań binarnych T o n węzłach, gdzie kluczami w węzłach będą symbole ze zbioru S , takie aby średni czas zlokalizowania klucza w drzewie był minimalny (drzewo takie jest nazywane *optymalnym drzewem BST*).

Wskazówka. Zminimalizuj wartość $\sum_{v \in T} d_v p_v$, gdzie d_v jest głębokością węzła v (zakładamy, że korzeń leży na głębokości 1) a p_v jest prawdopodobieństwem z jakim występuje symbol zapamiętany w tym węźle.