Algorytmy i Struktury Danych

IIUWr. II rok informatyki.

1. (1pkt) Ułóż algorytm znajdujący najtańszą drogę przejścia przez tablicę, w którym oprócz ruchów dopuszczalnych w wersji problemu prezentowanej na wykładzie, dozwolone są także ruchy w górę i w dół tablicy.

2. (1.5pkt) Ułóż algorytm, który dla danego ciągu znajduje długość najdłuższego jego podciągu, który jest palindromem.

3. (1.5pkt) Chcemy obliczać wartości współczynników dwumianowych modulo liczba pierwsza. Ułóż algorytm, który dla danej liczby pierwszej p oraz ciągu par $(n_1, k_1), (n_2, k_2), ...(n_r, k_r)$ obliczy wartości $\binom{n_i}{k_i} \bmod p$. Twój algorytm powinien wypisywać wartości $\binom{n_i}{k_i} \bmod p$ po czasie nie większym niż $O(\max\{n_1, n_2, \ldots, n_i\})$.

   Możesz założyć, że $k_i \leq n_i < p$ dla każdego $i = 1, \ldots, r$.

4. (2pkt) Zmodyfikuj algorytm znajdujący najdłuższy wspólny podciąg dwóch ciągów n elementowych, tak by działał w czasie $O(n^2)$ i używał $O(n)$ pamięci.

5. (2pkt) Dany jest graf pełny $G = (V, E)$ z nieujemnymi wagami na krawędziach oraz ciąg wszystkich jego wierzchołków $C = \langle v_1, \ldots, v_n \rangle$. Początkowo w wierzchołku v_1 znajdują się dwa pionki. W kolejnych ruchach masz przesunąć pionki według następujących zasad:

   - w każdym ruchu przesuwasz jeden pionek,
   - pionek stojący w wierzchołku v_i możesz być przesunąć do wierzchołka v_j jedynie wtedy, gdy $j > i$ (czyli do wierzchołka znajdującego się dalej w ciągu C),
   - wszystkie wierzchołki grafu muszą być odwiedzone przez co najmniej jeden pionek,
   - po ostatnim ruchu obydwa pionki znajdują się w wierzchołku v_n.

   Ułóż algorytm obliczający ciąg ruchów pionków minimalizujący sumę długości dróg przebytych przez pionki (przez długość drogi rozumiemy sumę wag jej krawędzi).

6. (2pkt) Ułóż algorytmy, które dla danych podciągów x i y rozwiązują następujące wersje problemu znajdowania najdłuższego wspólnego podciągu:

   - znajdowanie najdłuższego wspólnego podciągu zawierającego podciąg ”matura”,
   - znajdowanie najdłuższego wspólnego podciągu nie zawierającego podciągu ”matura”,
   - znajdowanie najdłuższego wspólnego podciągu zawierającego podsłowo ”matura”,
   - znajdowanie najdłuższego wspólnego podciągu nie zawierającego podsłowa ”matura”.

7. (1pkt) Rozważmy następujący problem *3-podziału*. Dla danych liczb całkowitych $a_1, \ldots, a_n \in \langle -C..C \rangle$ chcemy stwierdzić, czy można podzielić zbiór $\{1, 2, \ldots, n\}$ na trzy rozłączne podzbiory I, J, K, takie, że

   $$\sum_{i \in I} a_i = \sum_{j \in J} a_j = \sum_{k \in K} a_k.$$

8. (2pkt) Dwie proste równoległe l' i l'' przecięto n prostymi $p_1, \ldots, p_n$. Punkty przecięcia prostej p_i z prostymi l' i l'' wyznaczają na niej odcinek. Niech *Odc* będzie zbiorem tych odcinków.

   (a) Ułóż algorytm, wyznaczający w *Odc* podzbiór nieprzecinających się odcinków, o największej mocy.

(b) Ułóż algorytm, wyznaczający liczbę podzbiorów, o których mowa w poprzednim punkcie.

9. (2pkt) Ułóż algorytm rozwiązujący poniższy problem triangulacji wielokąta wypukłego:

PROBLEM:

Dane: ciąg par liczb rzeczywistych $(x_1, y_1), \ldots, (x_n, y_n)$, określających kolejne wierzchołki n-kąta wypukłego P

ZAŁOŻENIE: dane są określone poprawnie.

Zadanie: Znaleźć zbiór S nieprzecinających się przekątnych, które dzielą P na trójkąty, taki, że długość najdłuższej przekątnej w S jest możliwie najmniejsza.

10. (**Z** 2pkt) Mamy dane dwa ciągi liczb $A[1..n]$ oraz $B[1..n]$. *Wspólnym rosnącym podciągiem* długości k ciągów A oraz B nazywamy ciąg indeksów $x_1 < x_2 < \ldots x_k$ oraz $y_1 < y_2 < \ldots y_k$ o następujących własnościach:

(a) $A[x_i] = B[y_i]$ dla każdego i,

(b) $A[x_1] < A[x_2] < \ldots < A[x_k]$.

Skonstruuj efektywny algorytm wyznaczający długość najdłuższego wspólnego rosnącego podciągu.

Czy potrafisz zmodyfikować Twój algorytm tak, aby używał tylko liniowej pamięci?