ASD 2022 – egzamin
20.02.2023

Zadanie 1 [13 punktów]
Niech n będzie liczbą całkowitą większą od 3. *Wachlarzem* W_n nazywamy nieskierowany graf o zbiorze n+1 wierzchołków {0, 1, 2, …, n} i zbiorze krawędzi {0—1, 0—2, …, 0—n, 1—2, 2—3, …, n-1 – n}.
Niech w: $E(W_n) \rightarrow Z^+$ będzie funkcją, która przypisuje krawędziom wachlarza W_n dodatnie wagi całkowitoliczbowe.

a) **[3 punkty]** Jaka może być najmniejsza a jaka największa wysokość drzewa przeszukiwania w głąb o korzeniu w wierzchołku 1 (mierzona liczbą krawędzi) w wachlarzu W_n?
b) **[5 punktów]** Zaprojektuj wydajny algorytm, które obliczy drzewo przeszukiwania w głąb o korzeniu w wierzchołku 1 o minimalnej sumie wag krawędzi.
c) **[5 punktów]** Zaproponuj wydajny algorytm obliczający dla wszystkich wierzchołków w W_n wagi najlżejszych ścieżek łączących te wierzchołki ze źródłem 0.

Zadanie 2 [13 punktów]
W tym zadaniu rozważamy kwadraty jednostkowe na płaszczyźnie kartezjańskiej, których rogi mają współrzędne całkowitoliczbowe. Kwadrat, którego dolny lewy róg ma współrzędne (x,y) oznaczamy przez $K(x,y)$. Powiemy, że dwa kwadraty są *sąsiednie* wtedy i tylko wtedy, gdy mają wspólny bok. Dany jest skończony zbiór kwadratów S. Dwa kwadraty A i B ze zbioru S są *spokrewnione* wtedy i tylko wtedy, gdy istnieje w S ciąg kwadratów $K_0 = A$, K_1, …, $K_i = B$ takich, że dla każdego j = 0, …, i-1, K_j sąsiaduje z K_{j+1}. Przyjmujemy, że kwadrat jest spokrewniony sam z sobą. Każdy maksymalny podzbiór spokrewnionych kwadratów w S nazywamy *składową S*.

a) [8 punktów]
Zaproponuj strukturę danych umożliwiającą wydajne wykonywanie operacji na dynamicznym skończonym zbiorze S:
[4 punkty]
Ini(S):: S := Θ; //operacja wykonywana tylko raz na samym początku;
Search(x,y):: sprawdź, czy kwadrat $K(x,y)$ jest w S;
Insert(x,y):: dodaj do zbioru S kwadrat $K(x,y)$;
[4 punkty]
Count(x,y):: policz ile kwadratów w zbiorze S jest spokrewnionych z kwadratem $K(x,y)$, o ile taki kwadrat jest w S

b) [5 punktów]
Załóżmy, że dany jest zbiór S złożony z n kwadratów umieszczonych w kwadracie o dolnym lewym roku w punkcie (0,0) i o boku długości n^2. Zaprojektuj wydajny algorytm, który obliczy rozmiar największej składowej w S.

Zadanie 3 [10 punktów]

W tym zadaniu rozważamy słowa zbudowane nad alfabetem {d ≤ i ≤ k ≤ s}. Powiemy, że słowo *t*[1..*k*] jest uporządkowane wtedy i tylko wtedy, gdy dla każdego *i* = 1, 2, …, *k*-1, *t*[*i*] ≤ *t*[*i*+1].

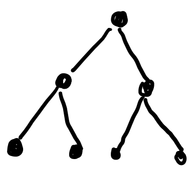
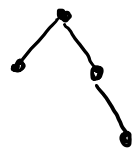
a) **[5 punktów]**
Zaproponuj wydajny algorytm, który oblicza ile jest różnych uporządkowanych podsłów w danym słowie *x*[1..*n*].

b) **[5 punktów]**
Zaproponuj wydajny algorytm, który oblicza ile jest różnych podsłów w danym słowie *x*[1..*n*], które występują w nim dokładnie raz.

Zadanie 4 [4 punktów]

AVL-drzewo o wysokości *h* z najmniejszą możliwie liczbą wierzchołków nazywamy *h*-minimalnym. Udowodnij, że dla każdego *h* ≥ 0, każde AVL-drzewo *T* o wysokości zawiera poddrzewo *h*-minimalne o tym samym korzeniu co *T*.

Przykład

Dla drzewa  h-minimalnym poddrzewem jest np. 

Uwaga: uzasadnij poprawność swoich rozwiązań oraz dokonaj analizy złożoności obliczeniowej zaproponowanych algorytmów.