ASD 2022 – egzamin
7.02.2023

Zadanie 1 [13 punktów]

a) **[5 punkty]** Niech G będzie nieskierowanym grafem spójnym z parami różnymi wagami na krawędziach (liczby całkowite) i niech e będzie krawędzią, dla której istnieje cykl elementarny ją zawierający i taki, że waga krawędzi e jest największa spośród wag wszystkich krawędzi na tym cyklu. Udowodnij, że e nie należy do minimalnego drzewa rozpinającego grafu G.

b) Niech n będzie liczbą całkowitą większą od 3. Wachlarzem W_n nazywamy nieskierowany graf o zbiorze 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 parami różne, całkowitoliczbowe wagi.
b1) **[2 punkty]** Jaka może być najmniejsza a jaka największa średnica minimalnego drzewa rozpinającego w wachlarzu W_n, w zależności od wag w?
B2) **[6 punktów]** Zaproponuj wydajny algorytm obliczający minimalne drzewo rozpinające w wachlarzu W_n.

Zadanie 2 [16 punktów]

Dane jest n-wierzchołkowe drzewo T = ({1,2, …, n}, F) z dodatnimi, całkowitoliczbowymi wagami na krawędziach.

a) **[4 punkty]** Zaproponuj wydajny algorytm, który oblicza wagę najcięższej ścieżki elementarnej (bez powtarzających się wierzchołków) w drzewie T.

b) **[4 punkty]** Zaproponuj wydajny algorytm, który dla drzewa T wyznaczy najlżejszą marszrutę (ciąg kolejno odwiedzanych wierzchołków) odwiedzającą każdy wierzchołek drzewa co najmniej raz. Wagą marszruty jest suma wag krawędzi w tej marszrucie.

c) **[8 punktów]** Zaproponuj strukturę danych, która umożliwia wydajne wykonanie sekwencji m operacji z podanego poniżej zestawu, gdzie $m > n$:
Ini():: G := T; //operacja wykonywana tylko raz, na samym początku;
Add(a – b):: jeśli wierzchołki a, b są w relacji przodek-potomek w drzewie T **ukorzenionym w wierzchołku 1** i a – b nie jest krawędzią w G, dodaj krawędź a – b do grafu G;
BCs():: podaj ile jest dwuspójnych wierzchołkowo składowych w (aktualnym) grafie G.

Zadanie 3 [5 punktów]

Przedziałem parzystym/nieparzystym nazywamy każdy skończony ciąg kolejnych liczb parzystych/nieparzystych. Dla przykładu ciąg 2, 4, 6, 8 (5, 7, 9) jest przedziałem parzystym (nieparzystym), ale ciąg 2, 4, 8 (5, 9) już nie jest przedziałem parzystym (nieparzystym).
Zaproponuj strukturę danych, która umożliwia wydajne wykonywanie następujących operacji na dynamicznym skończony zbiorze S liczb całkowitych:
Ini(S):: S := Θ //tylko raz na samym początku
Insert(a):: S := S U {a}
Delete(a):: S := S \ {a}
LongestOdd():: podaj długość najdłuższego przedziału nieparzystego zbudowanego z elementów zbioru S
LongestEven():: podaj długość najdłuższego przedziału parzystego zbudowanego z elementów zbioru S

Zadanie 4 [6 punktów]

Powiemy, że skończony ciąg cyfr dziesiętnych c_1, c_2, …, c_k jest F-ciągiem wtedy i tylko wtedy, gdy $k > 2$ oraz dla każdego $i = 3, \ldots, k$, $c_i = (c_{i-2} + c_{i-1}) \bmod 10$. Dane jest skończone słowo *x* nad alfabetem {0, 1, 2, …, 9}. Zaproponuj wydajny algorytm, który oblicza liczbę wszystkich różnych podsłów słowa *x*, które są F-ciągami.

Uwaga: podsłowo słowa *x,* to każdy spójny fragment tego słowa; te same podsłowa mogą zaczynać się w różnych miejscach słowa, np. słowo aaa zawiera trzy niepuste podsłowa: a, aa, aaa.