Egzamin z ASD
8.02.2022 r.

Zadanie 1 [16 punktów]

W tym zadaniu rozważamy skończone słowa binarne – słowa nad alfabetem {0,1}. Przez n oznaczamy długość słowa.

a) [**5 punktów**] Zaprojektuj wydajny algorytm, który dla danego niepustego słowa x obliczy liczbę wszystkich par indeksów (i,j), $1 \le i \le j \le |x|$, takich że podsłowo $x[i..j]$ jest zapisem binarnym liczby podzielnej przez 3.
Uwaga: tutaj dopuszczamy, żeby 0 było najbardziej znaczącą cyfrą w zapisie. Dla przykładu dla x = <0110> liczba takich par wynosi 6: (1,1), (1,3), (1,4), (2,3), (2,4), (4,4).
W tym zadaniu dopuszczamy wykonywanie w czasie stałym tylko operacji arytmetycznych na liczbach o długościach zapisu binarnego O(log n).

b) [**5 punktów**] Teraz przyjmijmy, że słowo x jest dynamiczne, początkowo puste. Zaproponuj strukturę danych, która umożliwi wydajne wykonywanie na słowie x następujących operacji:

Wstaw(i,b): wstaw bit b na pozycję i w słowie x, $1 \le i \le n+1$ ($x := x[1..i-1] \bullet b \bullet x[i..n]$;

Usuń(i): usuń i-ty bit ze słowa x, $1 \le i \le n$ ($x := x[1..i-1] \bullet x[i+1..n]$;

Podzielne3: podaj liczbę par indeksów (i,j), $1 \le i \le j \le |x|$, takich że podsłowo $x[i..j]$ jest zapisem binarnym liczby podzielnej przez 3.

c) [**6 punktów**] Powiemy, że zapis dodatniej liczby całkowitej jest krótki, jeśli najbardziej znaczący bit w tym zapisie jest jedynką. Zaprojektuj wydajny algorytm, który dla danego słowa binarnego x obliczy liczbę jego wszystkich parami różnych podsłów, które są krótkimi zapisami dodatnich liczb całkowitych podzielnych przez 4.

Zadanie 2 [12 punktów]

Dana jest dodatnia liczba całkowita n i tablica liczb całkowitych $a[1..n]$.

a) [**6 punktów**] Załóżmy, że elementami tablicy są tylko dwie różne liczby i każda z nich pojawia się w tablicy co najmniej raz. Zaproponuj wydajny algorytm, który znajdzie parę indeksów (i, j) o największej różnicy j-i, $1 \le i < j \le n$, taką że podtablica $a[i..j]$ zawiera każdą z tych dwóch liczb tyle samo razy.

b) [**6 punktów**] Załóżmy teraz, że elementami tablicy są trzy różne liczby i każda z nich pojawia się co najmniej raz. Zaproponuj wydajny algorytm, który znajdzie parę indeksów (i, j) o największej różnicy j-i, $1 \le i \le j \le n$, taką że podtablica $a[i..j]$ zawiera każdą z tych trzech liczb tyle samo razy. Jeśli taka para nie istnieje Twój algorytm jako wynik ma podać parę (0,0).

Zadanie 3 [5 punktów]

Dana jest dodatnia liczba całkowita n. Zaprojektuj strukturę danych, która na początkowo pustym multigrafie G o wierzchołkach 1, 2, …, n umożliwia wydajne wykonywanie następujących operacji:

Dodaj(u,v):: dodaj krawędź u—v do grafu G;

EulerCyc:: podaj liczbę spójnych składowych w grafie, które są eulerowskie (posiadają cykl Eulera).

Zadanie 4 [7 punktów]

Do początkowo pustego AVL-drzewa wstawiamy kolejno liczby 1, 2, 3, …, *n*. Węzły utożsamiamy z zapisanymi w nich kluczami. Ponieważ wiadomo, że *i* zostanie wstawiony jako prawe dziecko węzła *i-1*, to pomijamy w procedurze wstawiania fazę na poszukiwanie miejsca, w którym *i* ma się znaleźć – możemy go od razu podwiązać jako prawe dziecko *i-1*. Musimy jeszcze przywrócić własność bycia AVL-drzewem i zaktualizować wartości wskaźnika zrównoważenia w węzłach drzewa. W tym celu wykonujemy drugą fazę wstawiania do AVL-drzewa. Możesz założyć, że każdy węzeł w drzewie ma 5 atrybutów:
klucz, *lewy*, *prawy*, ojciec - wskaźniki odpowiednio do lewego i prawego dziecka oraz ojca, *wzr* – wskaźnik zrównoważenia.
Dokonaj analizy kosztu zamortyzowanego operacji wstawiania do AVL-drzewa w takim przypadku.

Uzasadnij poprawność swoich rozwiązań i dokonaj analizy złożoności obliczeniowej zaproponowanych algorytmów. Rozwiązanie każdego podzadania zapisz na oddzielnej kartce.