

ASD – klasówka 2

10.01.2025

Zadanie 1 [10 punktów]

W tym zadaniu rozważamy domknięte przedziały liczb całkowitych. Każdy przedział jest reprezentowany przez parę liczb całkowitych a, b , gdzie $a \leq b$ i a jest **początkiem** przedziału, natomiast b jego **końcem**. Skończony zbiór przedziałów S nazywamy **pełnym** wtedy i tylko wtedy, gdy żadne dwa przedziały z S nie mają wspólnego końca i każde dwa przedziały w tym zbiorze mają niepuste przecięcie.

- a) **[2 punkty]** Dany jest pełny zbiór S złożony z n przedziałów. Zaproponuj wydajny algorytm, który znajdzie przedział będący iloczynem teoriomnogościowym wszystkich przedziałów z S .
- b) **[4 punkty]** Zaproponuj strukturę danych dla dynamicznego, skończonego i pełnego zbioru przedziałów S umożliwiającą wydajne wykonywanie następujących operacji:
Ini(S):: $S := \emptyset$ //Operacja wykonywana tylko raz, na samy początku.
Add($S, [a, b]$):: jeśli $S \cup \{[a, b]\}$ jest pełny, to $S := S \cup \{[a, b]\}$
Remove($S, [a, b]$):: $S := S \setminus \{[a, b]\}$
Product(S):: **return** $\bigcap_{S \in S} S$;
- c) **[4 punkty]** Rozszerz zbiór operacji z b) o dwie następujące operacje:
In($S, [a, b]$):: sprawdź, czy w zbiorze S jest przedział całkowicie zawarty w $[a, b]$; jeśli tak, to podaj jeden z takich przedziałów
Remove2($S, [a, b]$):: jeśli b jest niemniejsze od największego z początków przedziałów z S , to usuń ze zbioru S wszystkie przedziały, które mają niepuste przecięcie z przedziałem $[a, b]$

Zadanie 2 [6 punktów]

- a) **[2 punkty]** Dane jest drzewo binarne T z kluczami całkowitoliczbowymi w węzłach. Zaprojektuj wydajny algorytm, który sprawdzi, czy T jest drzewem wyszukiwań binarnych.
- b) **[4 punkty]** Celem tego zadania jest zaprojektowanie struktury danych, która umożliwi wydajną symulację ciągu następujących operacji na początkowo pustym drzewie wyszukiwań binarnych T z kluczami całkowitoliczbowymi. Klucze utożsamiamy z węzłami, w których się znajdują w drzewie.
Search(x, T):: sprawdź, czy x jest w aktualnym drzewie T
Insert(x, T):: jeśli x nie ma w drzewie T , to dodaj x do drzewa
Leaf(x, T):: sprawdź, czy x jest liściem w T
DeleteLeaf(x, T):: jeśli x jest liściem w drzewie T , to usuń x z T
Depth(x, T):: jeśli x jest w T , to podaj głębokość x w tym drzewie
Subtree(x, T):: jeśli x jest w T , to podaj liczbę węzłów w poddrzewie T o korzeniu x

Zadanie 3 [4 punkty]

Dane są trzy stosy, które początkowo zawierają po jednym żetonie każdy. Następnie wykonujemy ciąg kroków. W każdym kroku do jednego, dowolnie wybranego stosu dokładamy jeden żeton. Jeśli wysokość najwyższego stosu stanie się dwukrotnie większa od sumy wysokości dwóch pozostałych stosów, to przekładamy połowę żetonów z najwyższego stosu na pozostałe dwa stosy tak, aby na wszystkich stosach było po tyle samo żetonów.

Operacjami jednostkowymi są położenie żetonu na stos i zdjęcie żetonu ze stosu.

Oblicz zamortyzowany koszt jednego kroku metodą funkcji potencjału.