

Ćwiczenia 16

- 12.1 Zaprojektuj strukturę danych, która umożliwia efektywne wykonywanie ciągu operacji $\text{Łącz}(u, v)$ i $\text{Głębokość}(u)$ na lesie drzew ukorzenionych o zbiorze wierzchołków $\{1, 2, \dots, n\}$. Początkowo każde drzewo jest jednowierzchołkowe. Operacja $\text{Łącz}(u, v)$ polega na połączeniu dwóch różnych drzew o korzeniach u i v w jedno drzewo o korzeniu v , poprzez uczynienie u synem v (podwiązanie u do v). Operacja $\text{Głębokość}(u)$ polega na wyznaczeniu głębokości wierzchołka u w aktualnie zawierającym go drzewie w lesie. Podaj sposób i koszt inicjacji swojej struktury danych, a następnie koszt wykonania każdej z operacji Łącz i Głębokość w zaprojektowanym przez siebie rozwiązaniu.

Przykład: Dla $n = 5$, po wykonaniu operacji $\text{Łącz}(1, 2)$, $\text{Łącz}(2, 3)$, $\text{Łącz}(3, 4)$, $\text{Łącz}(4, 5)$ wynikiem $\text{Głębokość}(2)$ jest 3.

- 12.2 Zaproponuj implementację struktury danych udostępniającej operacje struktury Find-Union dla elementów $1..n$ z przypisanymi całkowitoliczbowymi wartościami (początkowo same zera) oraz dwie nowe operacje:

- $\text{Add}(i, a)::$ do wartości wszystkich elementów ze zbioru zawierającego element i dodaj wartość a ,
- $\text{Value}(i)::$ podaj aktualną wartość przypisaną elementowi i .

- 12.3 Mamy n kul ponumerowanych od 1 do n . Na początku wszystkie kule są zielone. Na kulach wykonujemy następujące operacje:

- $\text{Pokoloruj}(a, b, kol):: 1 \leq a \leq b \leq n, kol \in \{\text{zielony}, \text{czerwony}\}$ — pokoloruj kule o numerach od a do b na kolor kol ,
- $\text{Kolor}(a):: 1 \leq a \leq n$ — podaj kolor kuli o numerze a .

- (a) Zaproponuj strukturę danych, która umożliwi efektywne wykonywanie ciągu operacji Pokoloruj i Kolor .
- (b) Załóżmy, że na początku wykonujemy $m \geq n$ z góry znanych operacji Pokoloruj , a następnie pytamy o kolor każdej kuli. Zaproponuj efektywny algorytm obliczający kolory kul po wykonaniu wszystkich operacji Pokoloruj .

- 12.4 Dana jest dodatnia liczba całkowita n oraz z góry zadany ciąg σ operacji $\text{Insert}(e)$ oraz ExtractMin , wykonywanych na początkowo pustym zbiorze S . Dla każdego $e \in \{1, 2, \dots, n\}$ operacja $\text{Insert}(e)$ jest wykonywana co najwyżej raz i polega na dodaniu e do zbioru S . Operacja ExtractMin znajduje w S element najmniejszy i usuwa go z tego zbioru. Możesz założyć, że ExtractMin jest zawsze dobrze określona. Zaproponuj wydajny algorytm, który poda wyniki wszystkich operacji ExtractMin .

- 12.5 Dokonaj analizy rozwiązania problemu Find-Union ze zrównoważaniem drzew i kompresją ścieżek, przy założeniu że operacje Find wykonywane są dopiero po wykonaniu wszystkich operacji Union.