
ALGORYTMY I STRUKTURY DANYCH

TABLICE DYNAMICZNE I ZBIORY ROZŁĄCZNE

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

-
1. *Licznik binarny* to struktura danych reprezentowana przez k -elementową tablicę bitów $B[0 \dots k - 1]$. W tablicy tej pamiętana jest nieujemna wartość całkowita w postaci binarnej, zwana licznikiem. Na liczbie tej można wykonywać jedynie operację `increment()` zwiększającą wartość licznika o 1. Operacja ta zaprogramowana jest w oparciu o pisemne dodawanie jedynki do liczby binarnej:

```
(1)   procedure increment (bit  $B[0 \dots k - 1]$ )
(2)   {
(3)     for  $i = 0 \dots k - 1$  do
(4)       if  $B_i = 1$  then  $B_i \leftarrow 0$ ;
(5)       else
(6)         {
(7)            $B_i \leftarrow 1$ ;
(8)           return ;
(9)         }
(10)  }
```

Określ faktyczny czas działania pojedynczej operacji `increment()`. Przeanalizuj czas wykonania ciągu n takich operacji na początkowo wyzerowanym liczniku.

- (a) [*] Dokonaj analizy zamortyzowanej *metodą kosztu sumarycznego*.
- (b) [*] Dokonaj analizy zamortyzowanej *metodą księgowania*.
- (c) [*] Dokonaj analizy zamortyzowanej *metodą potencjału*.
2. [*] Na pewnej strukturze danych wykonujemy ciąg n operacji. Rzeczywisty koszt i -tej operacji wynosi i , jeśli i jest potęgą 2, a w przeciwnym przypadku wynosi 1. Dokonaj analizy kosztu zamortyzowanego tych operacji *metodą kosztu sumarycznego*.
3. [**] Pokaż, jak zaimplementować *kolejkę* używając dwóch pomocniczych *stosów*. Określ czas działania pojedynczych operacji `append()` i `pop()` na takiej strukturze danych. Przeanalizuj czas wykonania ciągu n operacji na początkowo pustej kolejce. Dokonaj analizy zamortyzowanej *metodą księgowania*.

Czy w analogiczny sposób można zaimplementować stos używając pomocniczych kolejek?

4. [***] Dokonaj zamortyzowanej analizy czasu działania ciągu n operacji `append()` i `cut()` na początkowo pustej tablicy dynamicznej. Współczynnik charakterystyczny dla tej tablicy wynosi $\alpha = \frac{1}{2}$. Przeprowadź analizę kosztów *metodą potencjału*.
5. [***] Jak zaimplementować operacje `append()` i `cut()` na tablicy dynamicznej, aby w każdym przypadku czas ich działania był stały? Jakie wówczas będzie zużycie pamięci w pesymistycznym przypadku?
6. [*] Dany jest graf nieskierowany $G = (V, E)$ złożony z k spójnych składowych. Dla grafu tego wywołano następującą procedurę obliczającą liczbę składowych spójności:

```

(1)   function conected-components (graph  $G$ )  $\mapsto$  integer
(2)   {
(3)     DisjointSets  $S$ ;
(4)      $S.init(G.V)$ ;
(5)     for each  $(u, v) \in G.E$  do
(6)       if  $S.find(u) \neq S.find(v)$  then  $S.union(u, v)$ ;
(7)     return  $S.count-sets()$ ;
(8)   }
```

Ile razy nastąpi wywołanie `find()` a ile razy `union()`? Odpowiedź uzależnij od wielkości $|V|$, $|E|$ oraz k .

7. [**] Dany jest graf nieskierowany $G = (V, E)$ z ważonymi krawędziami. Jak za pomocą operacji na zbiorach rozłącznych wyznaczyć minimalne drzewo rozpinające grafu? Omów algorytm rozwiązujący to zadanie i przeanalizuj jego złożoność. Ile razy nastąpi wywołanie `find()` a ile razy `union()`? Odpowiedź uzależnij od wielkości $|V|$ i $|E|$.
8. [**] Zaproponuj efektywną strukturę danych do wykonywania ciągów podanych operacji na dynamicznie zmieniającym się podziale zbioru $\{1, 2, \dots, n\}$:
 - `init()` — utworzenie podziału $\{\{1\}, \{2\}, \dots, \{n\}\}$;
 - `union(A,B)` — połączenie zbiorów A i B bieżącego podziału w jeden;
 - `find(x)` — wyznaczenie zbioru, do którego należy element x ;
 - `find-smallest()` — wyznaczenie zbioru bieżącego podziału, który ma najmniej elementów.

Jaka jest złożoność tych operacji w twoim rozwiązaniu?

9. [**] Zaimplementuj operacje `union()` i `find()` dla drzewiastej reprezentacji zbiorów rozłącznych ze zbalansowanym łączeniem i kompresją ścieżek.
10. [**] Dana jest rodzina n jednoelementowych zbiorów rozłącznych. Zbiory te będziemy pamiętali w postaci drzewiastej ze zbalansowanym łączeniem. Skonstruuj ciąg $n-1$ operacji `Union`, by w ich wyniku powstał jeden zbiór reprezentowany przez n elementowe drzewo o wysokości $\lfloor \log n \rfloor$.
11. [***] Niech dany będzie ciąg n instrukcji `union()` i `find()`, w którym wszystkie instrukcje `union()` występują przed instrukcjami `find()`. Udowodnij, że koszt czasowy wykonania tych operacji na drzewiastych strukturach reprezentujących zbiory rozłączne ze zbalansowanym łączeniem i kompresją ścieżek wynosi $\Theta(n)$.