Karol Cidyło

Zadanie 9 z listy 2.

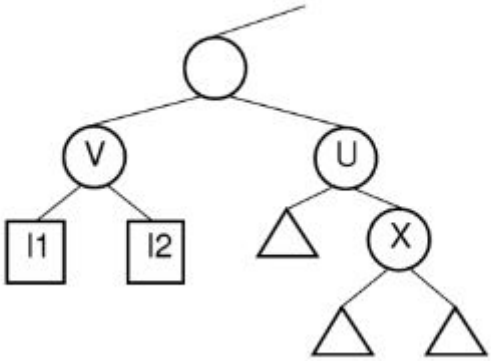
Algorytm
Drzewa Huffmana:

1. Stwórz n wierzchołków dla każdej wagi ze zbioru W $\{w_1, ... , w_n\}$, który jest zbiorem dodatnich liczb rzeczywistych.
2. Wstaw wszystkie wierzchołki do kolejki priorytetowej (porównując ze sobą wagi wierzchołków). Wierzchołek o najmniejszej wadze będzie korzeniem.
3. Powtarzaj dopóki kolejka priorytetowa zawiera więcej niż jeden element:
   (a) usuń dwa wierzchołki o najmniejszej wadze z kolejki - min_1 min_2
   (b) stwórz nowy wierzchołek, którego dziećmi są min_1 oraz min_2, a waga jest sumą wag min_1 oraz min_2
   (c) dodaj nowo utworzony wierzchołek w odpowiednie miejsce w kolejce
4. Zwróć ostatni wierzchołek, który pozostał w kolejce, jest on korzeniem drzewa.

Złożoność pamięciowa O(n). Tworzymy drzewo binarne. Pełne drzewo binarne z n liśćmi ma 2*n - 1 wierzchołków.
Złożoność czasowa O(n logn). Z każdym krokiem(3) zmniejszamy pierwotną kolejkę priorytetową o 1 element. Usuwanie oraz wstawianie do kolejki priorytetowej zajmuje log n. Znajdowanie minimum jest w czasie stałym. Razem mamy O(n logn).

Lemat.
Liście są rodzeństwem, kiedy mają tego samego rodzica(w drzewie binarnym oczywiście może być maksymalnie para rodzeństwa).
Dla każdego drzewa Huffmana zbudowanego przez powyższy algorytm zawierającego co najmniej dwa liście, te dwa liście z najmniejszą wagą są rodzeństwem i głębokość tych liści jest co najmniej taka sama jak każdego innego liścia w drzewie.

Dowód lematu:
Weźmy dwa liście z najmniejszymi wagami l_1 oraz l_2. Muszą być sąsiadami, ponieważ algorytm wybiera je na początku i tworzy z nich nowe drzewo. Załóżmy, że l_1 oraz l_2 nie są liśćmi na największej głębokości w drzewie. Niech V będzie rodzicem l_1 oraz l_2, wtedy musi istnieć inny wierzchołek X, który ma mniejszą wagę od wierzchołka V i jest on dzieckiem U (Rysunek 1). Gdyby tak nie było to algorytm wybrałby X zamiast V jako dziecko. Jednak jest to niemożliwe, ponieważ l_1 oraz l_2 są najmniejszymi wartościami dla wagi liści i ich suma też musi być najmniejsza ze wszystkich sum kombinacji duwelementowych ze zbioru W, ponieważ W $\{w_1, ... , w_n\}$ jest zbiorem dodatnich liczb rzeczywistych.



Rysunek 1: Nieprawidłowe drzewo Huffmana.[1]

Dowód indukcyjny:

- Podstawa: n = 2, drzewo Huffmana musi mieć minimalną zewnętrzną długość, ponieważ istnieją tylko dwa możliwe drzewa, w obu z nich dwa liście są na tej samej głębokości, więc mają również taką samą zewnętrzną długość.

- Założenie indukcyjne.
  Załóżmy, że każde drzewo zbudowane za pomocą algorytmu Huffmana zawierające n - 1 liści ma minimalną zewnętrzną długość.

- Krok indukcyjny:
  Weźmy drzewo T zbudowane za pomocą algorytmu Huffmana z n liśćmi, takie, że $n >= 2$.
  Bez straty ogólności możemy założyć, że $w_1 <= w_2 <= ...w_n$.
  Nazwijmy jako V rodzica dwóch pierwszych liści z wagami w_1 oraz w_2. Z lematu wiemy, że są one rodzeństwem i żaden inny liść z T nie jest na większej głębokości. Jeśli jakiś liść w drzewie byłby niżej moglibyśmy zmniejszyć zewnętrzną długość drzewa zamieniając z w_1 lub w_2. Ale ten lemat mówi, że nie ma takich liści.
  Weźmy drzewo T' identyczne z T z wyjątkiem wierzchołka V, który jest zastąpiony przez V' i jego waga jest równa $w_1 + w_2$. Z założenia indukcyjnego wiemy, że T' ma minimalną zewnętrzną długość, ponieważ ma n - 1 wierzchołków. Dodając dzieci V' do drzewa powstaje nam znowu drzewo T, więc musi mieć ono minimalną zewnętrzną długość.

□

[1]. Źródłem rysunku oraz informacji o algorytmie jest:
Data Structures & Algorithm Analysis in C++ Third Edition,
Clifford A. Shaffer.