AiSD ćwiczenia lista 2

Oskar Bujacz

24 kwietnia 2020

Zadanie 9

Dla ważonego drzewa $T = (V, E, c)$, gdzie $C : V \to \mathbb{R}_+$, określamy jego *zewnętrzną długość EL(T)* jako:

$$EL(T) = \sum_{v-\text{liść} \in T} c(v) \cdot d(v)$$

gdzie $d(v)$ jest długością ścieżki od korzenia do liścia v (mierzona liczbą krawędzi na ścieżce). Rozważmy następujący problem. Dany jest n-elementowy zbiór $\{w_1, \ldots, w_n\}$ dodatnich liczb rzeczywistych. Zadaniem jest znalezienie ważonego drzewa binarnego T o n liściach, takiego, że każda liczba w_i jest wagą dokładnie jednego liścia oraz T ma minimalną wagę $EL(T)$ pośród wszystkich drzew o tej własności.

1. Algorytm

W algorytmie będziemy korzystać z implemntacji kolejki priorytetowej poprzez kopiec. Oznaczmy przez L zbiór liści. Drzewem optymalnym będziemy nazywać drzewo binarne o minimalnej wartości $EL(T)$ dla danego zbioru w.

```
Q = kolejka priorytetowa liści ze zbioru L
for i = 1 : n-1
        w = nowy węzeł
        x = popmin(Q)
        y = popmin(Q)
        left(w) = x
        right(w) = y
        c(w) = c(x) + c(y)
        push(Q,w)
end
return pop(Q)
```

Teraz wystarczy pokazać, że algorytm tworzy drzewo optymalne względem $EL(T)$.

2. Kilka własności

Na początek pokażemy poprawność kilku lematów.

Lemat 1. *Niech T będzie drzewem optymalnym, wtedy każdy wierzchołek ma albo dwóch synów, albo jest liściem*

Dowód nie wprost. Załóżmy, że drzewo T' dla zbioru liści w jest optymalne i posiada wierzchołek v, który posiada tylko jednego syna u. Skoro budujemy drzewo dla zbioru liści, to możemy usunąć v oraz przesunąc u w jego miejsce. Zatem ścieżka do liścia (liści, bo u może być poddrzewem o więcej niż 1 liściu) uległą skróceniu o 1, zatem $EL(T')$ się zmiejszyło, sprzeczność z optymalnością. W szczególności lemat ten mówi też, że istnieją dwa wierzchołki będące braćmi.

Lemat 2. *Weźmy v i u będące liściami o minimalnej wadze z w, wtedy istnieje drzewo optymalne, w którym v i u są braćmi (mają wspólnego ojca) oraz są położone najgłębiej w drzewie (najdalej od korzenia).*

Dowód Niech T będzie drzewem optymalnym, załóżmy, że u i v są braćmi (z lematu 1) umieszczonymi najgłębiej (najniżej) w drzewie. Weźmy dwa inne liście x i y i załóżmy następujące nierówności.

$$c(u) \leqslant c(v), \quad c(x) \leqslant c(y), \quad c(x) \leqslant c(u), \quad c(y) \leqslant c(v)$$

Stwórzmy drzewo T' przez modyfikację drzewa T. Zamieńmy miejscami liście u i x. Wtedy

$$EL(T) - EL(T') = \sum_{v-\text{liść}\in T} c(v)\cdot d_T(v) - \sum_{v-\text{liść}\in T'} c(v)\cdot d_{T'}(v) =$$

$$= c(u)\cdot d_T(u) + c(x)\cdot d_T(x) - c(u)\cdot d_{T'}(u) - c(x)\cdot d_{T'}(x) =$$

$$= c(u)\cdot d_T(u) + c(x)\cdot d_T(x) - c(u)\cdot d_T(x) - c(x)\cdot d_T(u) =$$

$$= (c(u) - c(x))\,(d_T(u) - d_T(x)) \geqslant 0$$

Taka nierówność zachodzi, ponieważ $c(u) \geqslant c(x)$ z założenia. podobnie $d_T(u) \geqslant d_T(x)$, ponieważ u jest najgłebiej w drzewie. Otrzymaliśmy, że $EL(T') \geqslant EL(T)$, zatem T' również jest drzewem optymalnym. Stwórzmy kolejne drzewo T'' przez zamianę liści v i y. Analogiczna nierówność.

$$EL(T') - EL(T'') = \sum_{v-\text{liść}\in T'} c(v)\cdot d_{T'}(v) - \sum_{v-\text{liść}\in T''} c(v)\cdot d_{T''}(v) =$$

$$= c(v)\cdot d_{T'}(v) + c(y)\cdot d_{T'}(y) - c(v)\cdot d_{T''}(v) - c(y)\cdot d_{T''}(y) =$$

$$= c(v)\cdot d_{T'}(v) + c(y)\cdot d_{T'}(y) - c(v)\cdot d_{T'}(y) - c(y)\cdot d_{T'}(v) =$$

$$= (c(v) - c(y))\,(d_{T'}v) - d_{T'}(y)) \geqslant 0$$

Zatem $EL(T') \geqslant EL(T'')$, skoro T' jest optymalne, to również T'' jest. Pokazaliśmy, że istnieje drzewo optymalne, takie, że minimalna para liści jest rodzeństwem

Lemat 3. *Niech v i u będą liściami o najmniejszej wadze z w, niech x będzie takim liściem, że $c(x) = c(v) + c(u)$ i weźmy $L' = L\backslash\{u, v\}\cup\{x\}$. Wtedy jeśli T' jest optymalne dla L', to T w którym u, v są synami x jest optymalne dla L.*

Dowód Zauważmy, że dla każdego liścia v w L' bez x $d_T(v) = d_{T'}(v)$. Wiemy też, że $d_T(u) = d_T(v) = d_{T'}(x) + 1$. Mamy

$$c(u)\cdot d_T(u) + c(v)\cdot d_T(v) = (c(u) + c(v))\cdot(d_{T'}(x) + 1) =$$

$$= c(u) + c(v) + (c(u) + c(v))d_{T'}(x) = c(u) + c(v) + c(x)\cdot d_{T'}(x)$$

Skoro jedyne różnice między T i T' to liście u, v oraz x to $EL(T) = EL(T') + c(u) + c(v)$. Wiemy, że T' jest optymalne, załózmy nie wprost, że T nie jest optymalne dla zbioru L. Zatem istnieje wtedy drzewo T'' dla L, które jest optymalne,

$$EL(T'') < EL(T)$$

Wprowadźmy modyfikację do T'', zastąpmy liście najlżejsze liście u i v przez x (z lematu 2), $c(x) = c(u) + c(v)$.

$$EL(T'') - c(u) - c(v) < EL(T) - c(u) - c(v) = EL(T')$$

Sprzeczność z optymalnością T'.

3. Dowód poprawności algorytmu

Indukcyjnie pokażemy, że algorytm znajduje drzewo optymalne.
Baza indukcji
Dla $n = 1$ oczywiste, zwrócimy jedyny element w kolejce.
Krok indukcyjny
Załóżmy, że algorytm tworzy poprawne drzewo dla każdego $k < n$. Pokażemy dla n. W wyniku działania algorytmu powstanie pewne drzewo T o n liściach. W pierwszym obrocie pętli for weźmiemy dwa liście u i v o minimalnej wadze (lemat 2) i połączmy je w liść x o wadze $c(u) + c(v) = c(x)$. W kolejce pozostanie $n-1$ elementów. Z założenia indukcyjnego, wiemy, że potrafimy stworzyć drzewo optymalne dla $n - 1$ liści. Korzystając z lematu 3 możemy ponownie rozbić x na u i v zachowując optymalność. Indukcyjnie pokazaliśmy, że algorytm jest poprawny - znajduje minimalne drzewo binarne.

4. Analiza złożoności czasowej

Na początek tworzymy kolejkę priorytetową Q opartą na kopcu, czyli czas wymagany do jej stworzenia to $O(n)$. Następnie mamy pojedynczą pętlę for uruchomioną $n-1$ razy z operacjami usuwania minimum z Q i wstawiania elementu do Q, każda z nich o złożoności $O(log(n))$. Zatem złożoność pętli to $O(nlog(n))$, jak i całego algorytmu.