

## 1 Lista 6, Zadanie 5

Mamy przedstawić algorytm zawiadomienia wszystkich wierzchołków ukorzonego drzewa w możliwie najkrótszym czasie. W zadaniu nie jest zaznaczone czy jest to drzewo binarne czy może 2-3-4 drzewo. Przykłady podam dla drzewa binarnego, natomiast algorytm będzie można zaimplementować również dla drzew 2-3-4. W zadaniu mamy, że wierzchołek może poinformować jedno swoje dziecko w jednej iteracji. Wyobraźmy sobie, że informacja w zadaniu to płyn dezynfekujący, każdy węzeł, w tym liście, musi ów płyn otrzymać, przy czym płyn jest przenoszony z rąk do rąk przez wcieranie, czyli w jednej iteracji każdy kto do tej pory miał ręce w płynie może nanieść płyn na ręce swojego dziecka.

### 1.1 Algorytm

Przez  $time(v)$  oznaczać będziemy czas potrzebny na rozesłanie płynu w poddrzewie zakorzenionym w  $v$ . Dla liści  $time$  wynosić będzie 0.

Przykład w ramach tłumaczenia:

Węzeł  $v$  ma synów  $s_1$  i  $s_2$ ,  $time(s_1) = 2$ ,  $time(s_2) = 10$ . Możemy najpierw wysłać płyn do  $s_1$ , wtedy  $s_1$  skończy rozsyłać płyn do swoich dzieci w 2 iterację,  $s_2$  skończy w  $10 + 1$ , wtedy  $time(v) = 11$ . Jeżeli wyślemy w odwrotnej kolejności to  $time(v) = 10$ . Żeby zminimalizować czas propagacji, musimy najpierw wybierać dzieci potrzebujące więcej czasu (Chodzi tu oczywiście o dzietność dzieci a nie ich niezdarność). Jak je wybrać?

Żeby ustalić  $time$  dla danego węzła  $v$  potrzebujemy znać kolejność wysyłania dla jego dzieci. Dzięki przechodzeniu Post-order węzeł jest rozważany jako ostatni węzeł w swoim poddrzewie, czego właśnie potrzebowaliśmy. Dzieci  $v$  posortujemy malejąco po ich czasie propagacji, oznaczmy tę posortowaną sekwencję  $sequence(v)$ . Należy dodać, że dla każdego węzła  $v$ ,  $time(v) = \max(time(s_i) + i : s_i \in sequence(v))$ .

Zatem:

```
for  $v \in tree$  do
    if  $empty(v.children)$  then
         $v.sequence = []$ 
         $v.time = 0$ 
    else
         $v.sequence = sortDesc(v.children)$ 
         $v.time = \max(time(s_i) + i : s_i \in sequence(v))$ 
    end
end
```

#### Algorithm 1: Algorytm obliczający kolejność propagacji

W ten sposób po przejściu algorytmu, każdy węzeł ma informacje o tym ile najmniej czasu zajmie mu umycie rąk swoich dzieci. Kiedy będziemy już rzeczywiście przesyłać sygnał po tym drzewie, będąc w węźle  $v$  należy wybierać dzieci według porządku malejącego.