

# Zadania domowe. Zestaw 3.3

Maciej Poleski

20 maja 2013

# 1

Zbudujemy najpierw na tej tablicy kopiec. Czas liniowy. Pozostaje teraz zbudować strukturę (dane na temat odległości od liści). Możemy to zrobić przeglądając drzewo w kolejności postorder (czas liniowy).

# 2

W tym rozwiązaniu potrzebna jest kolejka priorytetowa złączalna na krawędziach (jedna krawędź może się pojawić łącznie co najwyżej 2 razy we wszystkich instancjach kolejki). Operacją złączania nie tylko musi być wydajna ( $\log n$ ), ale również działać jak różnica symetryczna. Możemy zaimplementować tę kolejkę przy użyciu drzewa lewicowego. Wszystkie operacje działają jak w drzewie lewicowym z wyjątkiem operacji `min()`. Którą implementujemy w ten sposób:

```
// wszystkie wywołania funkcji dotyczą funkcji dostarczanych
// przez drzewo lewicowe
while true:
  a <- extract_min()
  b <- min()
  if a != b:
    insert(a)
    return a
  else
    extract_min()
```

Czyli w skrócie wyciągamy wszystkie pojawiające się zduplikowane elementy z drzewa aż do napotkania pierwszego nie zduplikowanego - jego pozostawiamy i zwracamy. Złożoność jest zamortyzowana logarytmiczna (jeżeli wszystkich krawędzi jest  $n$ , to w czasie całego algorytmu w drzewie pojawi się co najwyżej  $2n$  elementów. Każdy element jest jeden raz umieszczany kolejno i co najwyżej raz usuwany z niej ( $0 \leq 1$ ). To daje koszt logarytmiczny na jedno wywołanie funkcji +  $n \log n$  na wywołanie ciągu funkcji. Amortyzuje się do logarytmicznego). `extract_min()` nie wykorzystujemy.

Algorytm: Znajdujemy jakieś minimalne drzewo rozpinające np. algorytmem Kruskala. Ukorzeniamy to drzewo jakkolwiek i przechodzimy w kolejności postorder. Dla każdego przetwarzanego poddrzewa posiadamy wyżej opisaną kolejkę priorytetową zawierającą krawędzie wychodzące z danego poddrzewa (bez krawędzi `parent` z korzenia danego poddrzewa). Priorytetem jest waga krawędzi. Krawędź `parent` (łączy korzeń danego poddrzewa

z jego ojcem) możemy zastąpić najbliższą krawędzią z kolejki priorytetowej ( $\text{min}()$ ). Dla danej krawędzi odpowiedzią jest koszt wyjściowego MST + różnica wag krawędzi “parent” i wybranej krawędzi z kolejki priorytetowej. Każda krawędź MST jest “parentem” jakiegoś wierzchołka w ukorzenionym MST. Natomiast usunięcie krawędzi nie należącej do MST nic nie zmienia. Na podstawie kolejek priorytetowych dzieci możemy uzyskać kolejkę priorytetową rodzica po prostu łącząc wszystkie kolejki dzieci (implementacji robi dla nas różnicę symetryczną - usunie te krawędzie które łączą ze sobą poddrzewa rodzeństwa w efekcie zostaną tylko te krawędzie które wychodzą z dokładnie jednego z poddrzew) i dodając (czyli mergując z singletonem) krawędzie niedrzewowe wychodzące z rodzica.

Kruskal  $E \log V$ , reszta  $E \log V$  ( $E$  insertów,  $V$   $\text{min}()$ ,  $V$   $\text{merge}()$ )