# Algorytmy i Struktury Danych, 4. ćwiczenia

2019-10-23

## Spis treści



## 1 Dowód, że $n-1$ porównań jest potrzebne do znajdowania minimum

Weźmy algorytm, $A$, powiedzmy, za każdym razem, gdy porównuje on dwa elementy, to łączymy je krawędzią. Jeśli $A$ użył mniej niż $n-1$ porównań, to istnieją dwa elementy, które nie są ze sobą porównywalne.

## 2 Optymalne znajdowanie drugiego co wielkości elementu

- budujemy drzewo turniejowe (porównujemy sąsiednie elementy, dalej przechodzi wygrany) — ten krok zabiera $n-1$ porównań,
- niech $S$ zbiór elementów które przegrały z liderem, $|S| = \lceil \log n \rceil$
- wybierz lidera wśród elementów $S$ — ten krok zabiera $|S|-1 = \lceil \log n \rceil - 1$ porównań.
- razem $n + \lceil \log n \rceil - 2$

Dowód, że algorytm jest optymalny. Knuth, tom III, 5.3.3. strona 221.

# 3 Sortowanie liczb z zakresu $0..n^3$

SORT(A)

1: posortuj stabilnie ciąg $A$ wg $A[i] \mod n$
2: posortuj stabilnie ciąg $A$ wg $\lfloor A[i]/n \rfloor \mod n$
3: posortuj stabilnie ciąg $A$ wg $\lfloor A[i]/n^2 \rfloor \mod n$

# 4 Sortowanie słów o różnych długościach

## 4.1 Sortowanie słów o różnych długościach, pierwsza próba

Niech $S = \{W_1, \ldots, W_n\}$ zbiór słów do posortowania, niech $n_i$ oznacza długość słów $W_i$.

RADIXSORT2(G)

1: wyzeruj tablicę $L$,
2: **for all** $W \in S$ **do**
3: dodaj $W$ na koniec listy $L[n_i]$.
4: **end for**
5: niech $n = max\{n_i : i \in 1, \ldots, n\}$,
6: $S = L[n]$
7: **for all** $i \in n, \ldots, 1$ **do**
8: wyzeruj tablicę $A$,
9: **for all** $W \in S$ **do**
10: dodaj $W$ na koniec listy $A[W[i]]$
11: **end for**
12: $S$ = złączenie listy $L[i-1]$ i list z tablicy $A$ (w tej kolejności)
13: **end for**

Niestety powyższy algorytm ma złożoność $O(|\Sigma|n)$, a my potrzebujemy $O(|\Sigma| + n)$.

## 4.2 Sortowanie słów o różnych długościach

RADIXSORT3(G)

1: przygotuj zbiór par $P = \{(i, x) : W[j][i] = x\}$,
2: posortuj $P$
3: wyzeruj tablicę $L$,
4: **for all** $W \in S$ **do**
5: dodaj $W$ na koniec listy $L[n_i]$.
6: **end for**
7: niech $n = max\{n_i : i \in 1, \ldots, n\}$,
8: $S = L[n]$
9: **for all** $i \in n, \ldots, 1$ **do**
10: **for all** $(i, x) \in P$ **do**
11: $A[x] = nil$
12: **end for**
13: **for all** $W \in S$ **do**
14: dodaj $W$ na koniec listy $A[W[i]]$
15: **end for**
16: $S = L[i-1]$

17: **for all** $(i, x) \in P$ **do**
18: $S = S \cup A[x]$
19: **end for**
20: **end for**