Zad 3.

poniedziałek, 20 marca 2023 18:54

3. (1pkt) Danych jest n odcinków $I_j = \langle p_j, k_j \rangle$, leżących na osi OX, $j = 1, \ldots, n$. Ułóż algorytm znajdujący zbiór $S \subseteq \{I_1, \ldots, I_n\}$, nieprzecinających się odcinków, o największej mocy.

procedure largest-compatible (I[1..n]):

1. sortujemy I[1..n] rosnąco względem końców odcinków
2. S[1..n] ← lista zawierająca początki przedziałów w kolejności z posortowanego I[1..n]
3. $A \leftarrow \emptyset$
4. indeks ← 1
5. dopóki indeks ≤ n rób:
   - $[p_i, k_i] \leftarrow I[\text{indeks}]$
   - $A \leftarrow A \cup \{[p_i, k_i]\}$
   - dopóki indeks ≤ n oraz S[indeks] ≤ k_i rób:
     - indeks ← indeks + 1
6. zwróć A

złożoność: $O(n \log n)$

- dowód poprawności

Niech $A = \{[a_1, b_1], \ldots, [a_k, b_k]\}$ oznacza zbiór wyprodukowany przez algorytm i niech $O = \{[a_1', b_1'], \ldots, [a_m', b_m']\}$ oznacza optymalne rozwiązanie. Chcemy pokazać, że $k = m$

Lemat
$\forall i \le k \quad b_i \le b_i'$

1° $i = 1$, oczywiste, bo $b_1 = \min\{k_j : [p_j, k_j] \in I\}$

2° zakładamy, że $b_{i-1} \le b_{i-1}'$. pokażemy, że $b_i \le b_i'$.

wiemy, że $b_{i-1}' < a_i'$, zatem $b_{i-1} < a_i'$.

[illegible] jest zatem wśród „dobrych"

wiemy, że $b'_{i-1} < a'_i$, zatem $b_{i-1} < a_i$. odcinek $[a'_i, b'_i]$ jest zatem wśród „dobrych" odcinków. Wiemy jednak, że algorytm w pierwszej kolejności wybiera spośród „dobrych" odcinków ten, o najmniejszej wartości końca. zatem $b_i \le b'_i$ $\square$

dalej, zakładamy nie wprost, że $m > k$. z lematu wiemy, że $b_k \le b'_k$. Skoro $m > k$, to mamy odcinek $[a'_{k+1}, b'_{k+1}] \in O$. $a'_{k+1} > b'_k$, zatem jednocześnie $a'_{k+1} > b_k$. Widać zatem, że algorytm dołożyłby również $[a'_{k+1}, b'_{k+1}]$. sprzeczność, zatem $k = m$ $\square$