Problem 2. Danych jest n odcinków $I_j =< p_j, k_j >$, leżących na osi OX, $j = 1, ..., n$. Ułóż algorytm znajdujący zbiór $S \subseteq I_1, ..., I_n$, nieprzecinających się odcinków, o największej mocy.

Proposition 1. Algorytm
Niech K będzie zbiorem wszystkich odcinków I_j, posortowanych po drugiej współrzędnej k_j.
Niech kmax - koniec ostatniego przetworzonego odcinka.

1. $S = \emptyset$
2. $i = 1$
3. $kmax = k_1 - 1$
4. Dopóki $i \leq n$:
   Jeśli $K[i].p > kmax$:
   - Dodaj $K[i]$ do S
   - $kmax = K[i].k$

   i++
5. Zwróć S

Niech optymalne rozwiązanie A o p-elementach i niech B to wyniku algorytmu.

Lemma . Dla każdego $i \leq p$, $B[i]$ istnieje i nie kończy się dalej niż $A[i]$.

Proof. Dowód indukcyjny po i
Baza indukcji:
$i = 1$ - jako pierwszy nasz algorytm wziął ten odcinek, który kończy się najwcześniej.
Krok indukcji:
Zakładam, że założenie indukcyjne jest prawdziwe dla każdego $k \leq i$.
Weźmy $i + 1$-wsze odcinki z powyższych zbiorów tzn. $A[i + 1]$ i $B[i + 1]$.
Rozważmy wszystkie możliwości:
Przypadek 1: $A[i + 1].k < B[i + 1].k$:
Nasz algorytm na miejsce $B[i + 1]$ wybrał pierwszy nieprzecinający się z $B[i]$ odcinek z posortowanego po k_j zbioru odcinków, zatem najkrótszy możliwy odcinek.
Dodatkowo z założenia indukcyjnego mamy, że $B[i].k \leq A[i].k$, zatem dostępne dla A odcinki są podzbiorem odcinków dostępnych dla B, czyli algorytm mógłby wybrać odcinek $A[i + 1]$, ale za optymalniejszy uznał $B[i + 1]$.
Skoro nasz algorytm nie wybrał tego odcinka, to $A[i + 1].k \geq B[i + 1].k$, co jest sprzeczne z założeniem przypadku.
Przypadek 2: $A[i + 1].k >= B[i + 1].k$:
Zatem $B[i + 1]$ kończy się nie dalej niż $A[i + 1]$. Zgodne z założeniem indukcyjnym.

∎

Zatem udowodniliśmy, że nasze rozwiązanie B jest nie gorsze niż dowolne rozwiązanie optymalne.

Złożoność czasowa:
Posortowanie po drugiej współrzędnej: $O(nlogn)$.
Przejście pętli w kroku 4.: $O(n)$.
Dodawanie odcinka do zbioru: Czas stały.
$O(n) + O(nlogn) = O(nlogn)$.