

Lista 4

Zadanie 5

1 Problem

Potrzebujemy algorytmu liczącego ilość nieporządków w ciągu a_1, \dots, a_n gdzie nieporządek rozumiemy jako zależność między elementami a_i, a_j taką, że $i < j \wedge a_i > a_j$.

2 Concept

Dla każdego a_i musimy sprawdzić wszystkie $\forall_{j>i} a_j$ czy nie zachodzi $a_i > a_j$.

Moglibyśmy tutaj wykorzystać algorytm `InsertionSort` i zliczać ilość przesunięć każdego z elementów podczas szukania ich właściwej pozycji na sortowanej tablicy. Jednakże, możemy podejść do problemu bezpośrednio i użyć dwóch pętli gdzie jedna z nich jest zagnieżdżona w drugiej. W obu przypadkach mamy taką samą złożoność obliczeniową z perspektywy O . (Akurat tutaj poniższy algorytm ma korzystniejszą stałą c stojącą obok funkcji n^2 .)

3 Rozwiązanie

Algorytm:

```
nieporządki ← 0
dla i = 1, ..., n:
  dla j = (i+1), ..., n:
    jeśli  $a_i > a_j$  wówczas:
      nieporządki ← nieporządki + 1
zwróć nieporządki
```