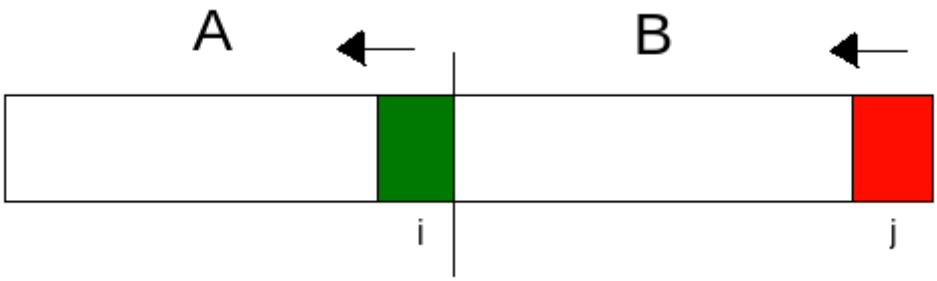
Lista 1 zadanie 6

Krystian Grabowski

Pomysł polega na podzieleniu tablicy na dwie równoliczne części(co do jednego elementu). Nazwijmy te części A i B. Następnie ustawiamy dwa wskaźniki. Pierwszy na koniec tablicy A, drugi na koniec tablicy B.



Następnie zaczynamy porównywać elementy znajdujące się na pod indeksami j oraz i. Jeśli $2*A[i] > B[j]$ to wiemy że $A[i]$ nie może zostać usunięty przez żaden inny element, gdyż $B[j]$ był elementem największym. Stąd możemy zmniejszyć i powtarzając proces. Jeśli natomiast $2*A[i] \leq B[j]$ to usuwamy te dwa elementy z ciągów.

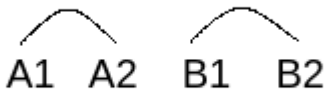
Nasze rozwiązanie działa w taki sposób że usuwa ono elementy z A2 B2 oraz A1 B1. Załóżmy, że istnieje rozwiązanie optymalne różne od naszego i usuwa ono elementy z A1 B2 i A2 B2. W optymalnym rozwiązaniu mamy następujące zależności: $$ 2*A1 \leq B2 $$ $$ 2*A2 \leq B1 $$

Z treści zadania wiemy, że ciąg jest ciągiem niemalejącym więc dla między zbiorami zachodzą zależności: $$ A1 \leq A2 \leq B1 \leq B2$$

Stąd: $$ 2*A2 \leq B1 \leq B2 $$ $$ 2*A2 \leq B2 $$

Oraz: $$ 2*A1 \leq 2*A2 $$ $$ 2*A2 \leq B1 $$ $$ 2*A1 \leq B1 $$

Rozwiązanie Optymalne może też wyglądać następująco:



Zachodzą w nim własności $$ 2*A1 \leq A2 $$ $$ 2*A2 \leq B1 $$ $$ 2*A1 \leq B2 $$

Możemy z nich wyprowadzić: $$ 2*A1 *leq A2 *leq B1 $$ $$ 2*A1 \leq B1 $$

Oraz $$ 2*A2 *leq 2*B1 \leq B2 $$ $$ 2*A2 \leq B2 $$

Więc jak widać możemy przekształcić rozwiązanie optymalne do naszego.

```
def delmax(array):
    half = len(array) // 2
    a, b = array[:half], array[half:]
    i = len(a) - 1
    j = len(b) - 1
    result = 0
    while (i != -1):
        if (2*a[i] <= b[j]):
            result += 2
            i -= 1
            j -= 1
        else:
            i -= 1
    return result
```