

Lab 4 Smol'ski, 323419, Zad 1.1

```
def w3g(X, n, k):  
    l = 0  
    i = 0  
    j = 0  
    p = 0  
    elem = 0  
    for i in range(n): elem = elem + X[i]  
    elem = elem / k  
    suma = X[0] - elem  
    if suma < 0: cdl = -suma  
    else: cdl = suma  
    while l != n-1:  
        if suma < 0:  
            if p == n-1: break  
            p = p + 1  
            suma = suma + X[p]  
            temp = suma  
            if temp < 0: temp = -temp  
            if temp < cdl:  
                cdl = temp  
                i = l  
                j = p  
        elif suma > 0:  
            if l == p:  
                suma = suma - X[l]  
                p = p + 1  
                l = l + 1  
            suma = suma + X[l]  
            temp = suma  
            if temp < 0: temp = -temp  
        if suma == 0:  
            print(i, j)  
    return 0
```

```
• ... if temp < cdl:  
    ..... cdl = temp  
    ..... i = l  
    ..... j = p  
• print(i, j)  
• return 0
```

c) W każdej iteracji
pętli dodajemy następny
z prawej element do sumy, jeśli
jest ona za mała, lub
~~nie~~ odejmujemy element
z lewej, gdy jest ona za
duża. Na koniec działania
programu zwracamy l i p,
dla których wartość (suma)
była najmniejsza, co jest
naszym rozwiązaniem.

b) Program iszcie limitów po tablicy
szukajże najmniejszego rozwiązania.
Zaczynamy od samego lewego elementu,
dodajemy element z prawej, jeśli suma jest
za mała, bierzemy z lewej, gdy jest
za duża. Jeśli w sumie jest tylko 1
element i suma jest za duża, to program
odejmuje ten element i dodaje następny.
Jeśli program znajdzie sumę równą c,
zakończy działanie. W innym przypadku
zakończy działanie, gdy nie może sprawdzić
wciąż liczb. Program

d) Wskaźniki l i p poruszają się
limitów po tablicy długości n.
W każdej iteracji pętli, u której
nie znaleźliśmy rozwiązania,
ciągłym przesuwamy się w prawo.
To daje nam złożoność $O(n)$