

Przykładowe zadanie 3 - etap powiatowy

Wstęp

Justyna Rojek-Nowosielska

Rozważmy następujący program, który analizuje ciąg liczb całkowitych dodatnich (możesz traktować je jako ceny pewnego towaru w kolejnych dniach).

C++

```
1 #include <iostream>
2 #include <vector>
3 #include <algorithm>
4 using namespace std;
5
6 int main() {
7     int n;
8     cin >> n;
9     vector<int> C(n);
10    for (int i = 0; i < n; i++) {
11        cin >> C[i];
12    }
13
14    int wynik = 0;
15
16    for (int j = 1; j < n; j++) {
17        for (int i = 0; i < j; i++) {
18            int roznica = C[j] - C[i];
19            if (roznica > wynik) {
20                wynik = roznica;
21            }
22        }
23    }
24
25    cout << wynik << endl;
26    return 0;
27 }
```

Program przyjmuje na wejściu liczbę n (liczba dni), a następnie n liczb całkowitych oznaczających wartości w kolejne dni.

? 1.1. Co oblicza powyższy program?

- (a) największą liczbę występującą w ciągu C ,
- (b) największą różnicę między dwoma sąsiednimi elementami ciągu,
- (c) największy możliwy zysk, kupując w dniu i i sprzedając w dniu j (gdzie $j > i$),
- (d) liczbę par (i, j) takich, że $C[j] > C[i]$.

? 1.2. Co wypisze program dla danych wejściowych: 5 oraz ciąg 7 1 5 3 6?

? 1.3. Podaj przykład 4-elementowego ciągu, dla którego program wypisze 0.

? 1.4. Ile operacji odejmowania wykona ten program dla $n = 100$? (Przybliżona wartość lub dokładny wzór).

? 1.5. Czy wynik działania programu może być liczbą ujemną przy założeniu, że wszystkie liczby na wejściu są dodatnie? Uzasadnij.

? 1.6. Jeśli na wejściu podamy ciąg posortowany rosnąco (np. 1, 2, 3, ..., 100), jaki będzie wynik?

- (a) 0
- (b) 100
- (c) 99
- (d) 5050

? 1.7. Ile jest par liczb (i, j) , które program sprawdza dla ciągu o długości $n = 5$?

? 1.8. Dla jakiego typu danych wejściowych (układ liczb) ten algorytm zwróci największą możliwą różnicę (równą różnicy między max i min całego ciągu)?

? 1.9. Zmodyfikuj (w wyobraźni) jedną linię kodu tak, aby program obliczał największą stratę (czyli największy spadek wartości między dniem i a j , gdzie $j > i$).

? 1.10. **Optymalizacja.**

Podany algorytm ma złożoność $O(n^2)$. Dla $n = 100\,000$ nie zakończy się w czasie 1 sekundy. Zauważ, że aby zmaksymalizować różnicę $C[j] - C[i]$ dla ustalonego j , musimy odjąć najmniejszą wartość $C[i]$ napotkaną **wcześniej**. Zaproponuj algorytm (słownie lub kod), który rozwiąże to zadanie, przeglądając tablicę tylko raz ($O(n)$).