

Rzeszów, 11.11.2021

## Sprawozdanie z projektu

Politechnika Rzeszowska  
Algorytmy i struktury danych

Krzysztof Motas  
Inżynieria i analiza danych, 1 rok, grupa nr 5

## Spis treści

|    |                                 |   |
|----|---------------------------------|---|
| 1. | Opis problemu .....             | 3 |
| 2. | Opis algorytmu .....            | 3 |
| 3. | Pseudokod .....                 | 3 |
| 4. | Schemat blokowy .....           | 4 |
| 5. | Wykres złożoności czasowej..... | 5 |

## 1. Opis problemu

Dla zadanej tablicy liczb całkowitych znajdź te pary, których różnica jest równa zadanej liczbie  $k$ .

## 2. Opis algorytmu

Rozpoczynamy od wprowadzenia do funkcji FindPairOfNumbers wektora przechowującego liczby całkowite oraz poszukiwanej wartości różnicy. Tworzymy mapę, w której będą przechowywane znalezione pary. Uruchamiamy dwie pętle za pomocą instrukcji iteracyjnej for, o zakresie długości liczby elementów podanego wektora. Porównujemy za pomocą instrukcji if wartość bezwzględną liczby z pierwszej pętli z liczbą z drugiej pętli. Dodatkowo musimy sprawdzić, czy różnica drugiej liczby i pierwszej liczby nie równa się szukanej wartości różnicy – ma to na celu wykluczenie powtarzających się przypadków, np. [2,5], [5,2]. Jeśli para liczb spełnia wyżej wymienione warunki, dodajemy ją do stworzonej mapy. Funkcja kończy się zwracaniem mapy, zawierającej znalezione pary liczb.

## 3. Pseudokod

```
map FindPairOfNumbers(vector array, diff)
```

```
{
```

```
    map foundPairs
```

```
    dla każdego i w array
```

```
        dla każdego x w array
```

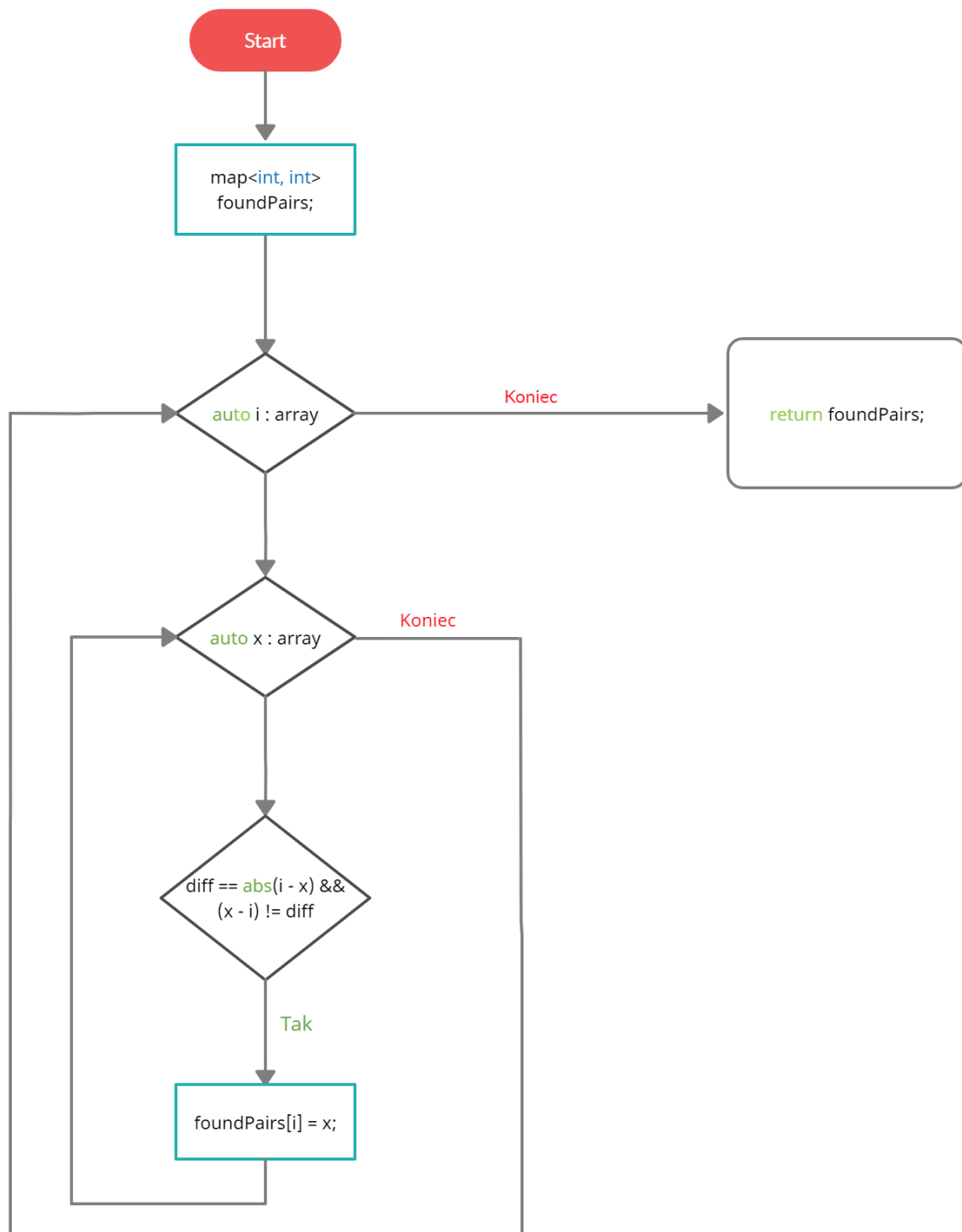
```
            jeżeli diff równa się wartości absolutnej  $(i - x)$  oraz  $(x - i)$  nierówna się diff wykonaj
```

```
                foundPairs[i] <- x
```

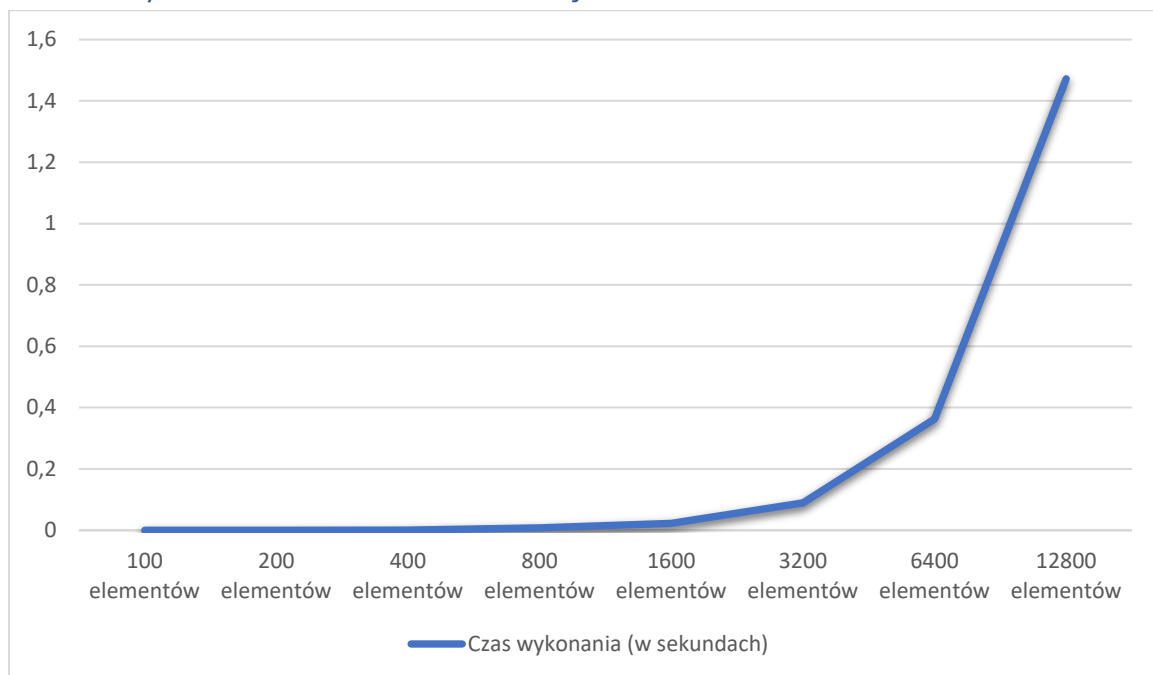
```
    zwróć foundPairs
```

```
}
```

#### 4. Schemat blokowy



## 5. Wykres złożoności czasowej



Wykres skonstruowany na podstawie wygenerowanych danych z funkcji:

```
void DoTimeMeasurements()
{
    clock_t time;

    fstream file;
    file.open("measurements.txt", ios::out);

    for (int i = 1, elements = 100; i < 9; i++)
    {
        time = clock();
        FindPairOfNumbers(GenerateRandomNumbers(1, 500, elements),
10);

        file << "Czas wykonania funkcji \"FindPairOfNumbers\" dla "
<< elements << " elementów wynosi ~" << (float)(clock() - time) /
CLOCKS_PER_SEC << " s.\n";
        elements *= 2;
    }

    file.close();
}
```