

Algorytmy i struktury danych

Projekt

P02 Kamil Bieniek

Kamil Bieniek

Inżynieria i analiza danych, 1. rok, grupa 1.

Porównanie 2 sposobów na filtrowanie danych liczb:

Sortowanie Kubełkowe polega na stworzeniu zmiennych o wartości występujących liczb a następnie do tej zmiennej zliczanie występowania tych liczb a następnie mnożenie tych liczników przez liczbe ich występowania

Przykład:

Wejście [12,11,15,16,18,14,15,14,19,11]

Liczniki: 10:0 11:2 12:1 13:0 14:2 15:2 16:1 17:0 18:1 19:1

Wyjście [11,11,12,14,14,15,15,16,18,19]

Sortowanie Przez Wybieranie polega przypisaniu pierwszej liczby do zmiennej a następnie sprawdzenie czy kolejna jest mniejsza ,jeśli tak to zmienia je pozycjami i przypisuje do zmiennej większą a następnie patrzy dalej

Przykład:

Wejście [12,11,15,16,18,14,15,14,19,11]

Zmienna =12 czy 11 < zmienna ,tak to zmienia pozycje

[11,12,15,16,18,14,15,14,19,11]

Zmienna =12 czy 15 < zmienna ,nie to zostawia bierze większą i szuka dalej

[11,12,15,16,18,14,15,14,19,11]

Wyjście [11,11,12,14,14,15,15,16,18,19]

Użyte biblioteki

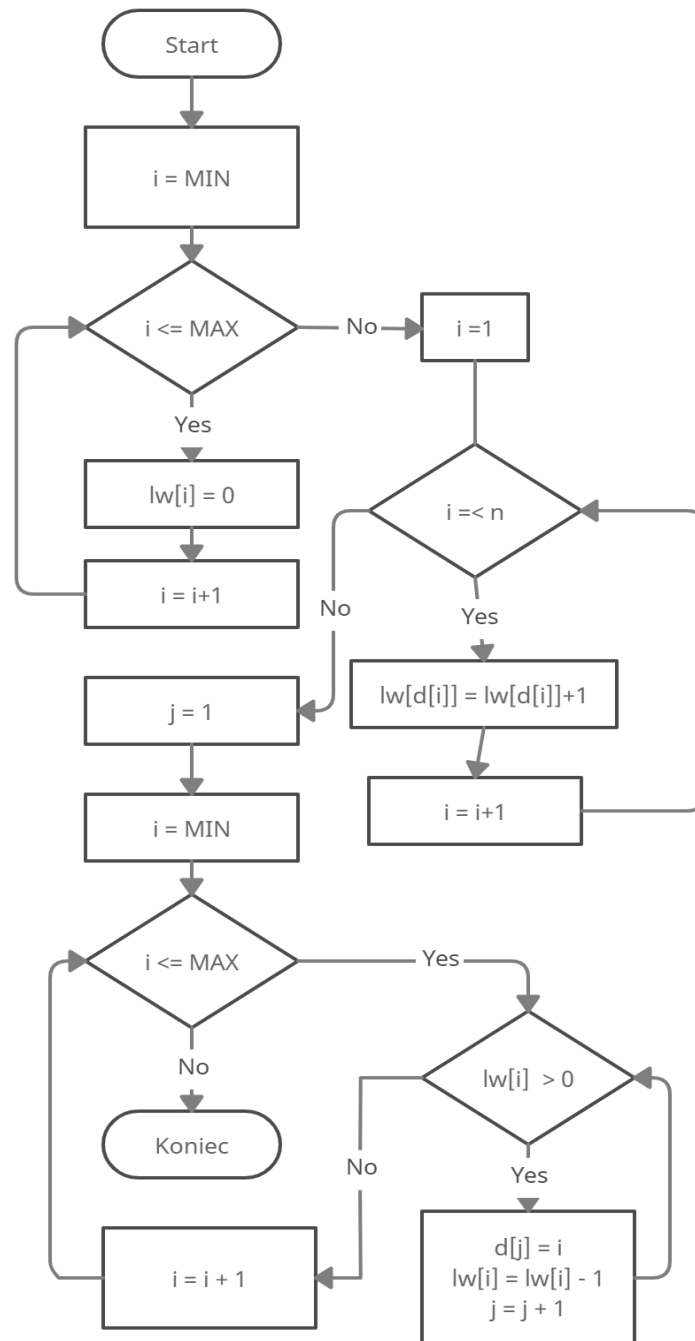
iostream	Biblioteka we-wyjścia. Deklaruje obiekty, które kontrolują odczytywanie ze strumieni standardowych i zapisywanie ich w tych strumieniach. Jest to często jedyny nagłówek potrzebny do wprowadzania danych i danych wyjściowych.
time.h	Udostępnia kilka typów danych, dzięki którym możemy odczytywać czas i wykonywać proste operacje na czasie, takie jak dodawanie czy odejmowanie. W przypadku tego zadania przydatna do generowania liczb pseudolosowych.
fstream	Dostarcza funkcji pozwalających nam zarówno zapisywać pliki jak i je odczytywać.
chrono	Definiuje klasy i funkcje, które reprezentują czasy trwania i czasy natychmiastowe. Tutaj – potrzebna przy liczeniu czasu działania programu.
omanip	Definiuje manipulatory z których każdy ma jeden argument

Użyte funkcje

Main	główna funkcja programu, zawiera inicjalizację i deklaracje zmiennych, funkcje związane z obliczaniem czasu obliczeń oraz zapisywaniem danych do pliku; w niej wywoływane są pozostałe funkcje
sortowanie_kubelkowe	Losuje liczby z podanego zakresu a następnie przeprowadza sortowanie kubelkowe
sorto_wybor	Losuje liczby z podanego zakresu a następnie przeprowadza sortowanie przez wybór

Tabela 1

Schemat Blokowy pierwszego sortowania



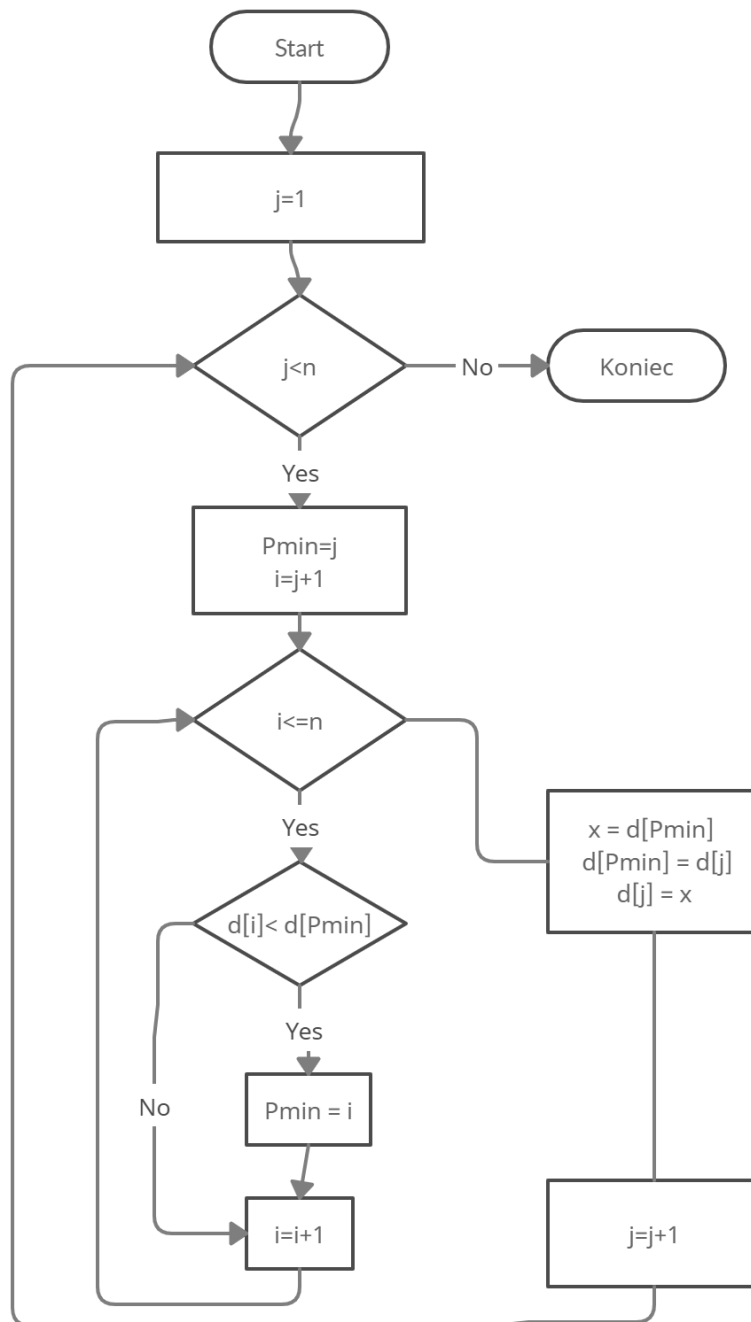
Pseudokod

1. For $l \leftarrow \text{MIN}$ to MAX
2. Do $i \leftarrow i+1$
3. $\text{lw}[i-\text{MIN}] \leftarrow 0$
4. For $l \leftarrow 0$ to n
5. Do $i \leftarrow i+1$
6. $\text{lw}[d[i] - \text{MIN}] \leftarrow \text{lw}[d[i] - \text{MIN}] + 1$
7. $j \leftarrow 0$
8. For $l \leftarrow \text{min}$ to MAX
9. Do $i \leftarrow i+1$
10. While $\text{lw}[i - \text{MIN}] \leftarrow \text{lw}[i - \text{MIN}] - 1$
11. Do $d[j \leftarrow j+1] = i$
- 12.
13. For $l \leftarrow \text{MIN}$ to n
14. Do $i \leftarrow i+1$
15. Wypisz $d[i]$

Wykonanie w oknie konsoli

```
1 sortowanie - sortowanie kubelkowe  
  
liczby przed sortowaniem:  
1 14 8 29 29 12 24 15 27 20 18 22 3 21 28 3 11 25 14 14  
Po sortowaniu:  
1 3 3 8 11 12 14 14 14 15 18 20 21 22 24 25 27 28 29 29  
  
Czas wykonania algorytmu w mikrosekundach:  
3958
```

Schemat blokowy drugiego sortowania



Pseudokod

1. For $j \leftarrow 0$ to $N - 1$
2. Do $j \leftarrow j + 1$
3. $pmin \leftarrow j$
4. For $i \leftarrow j + 1$ to N
5. Do $i \leftarrow i + 1$
6. If ($d[i] < d[pmin]$)
7. Do $pmin \leftarrow i$
8. Zamiana $d[pmin]$ i $d[j]$
9. For $i \leftarrow 0$ to N
10. Do $i \leftarrow i + 1$
11. Wypisz $d[i]$

Teraz 2 sortowanie - sortowanie przez wybieranie

14 6 15 14 10 28 30 16 11 21 14 22 2 21 15 10 17 25 24 13

Po sortowaniu:

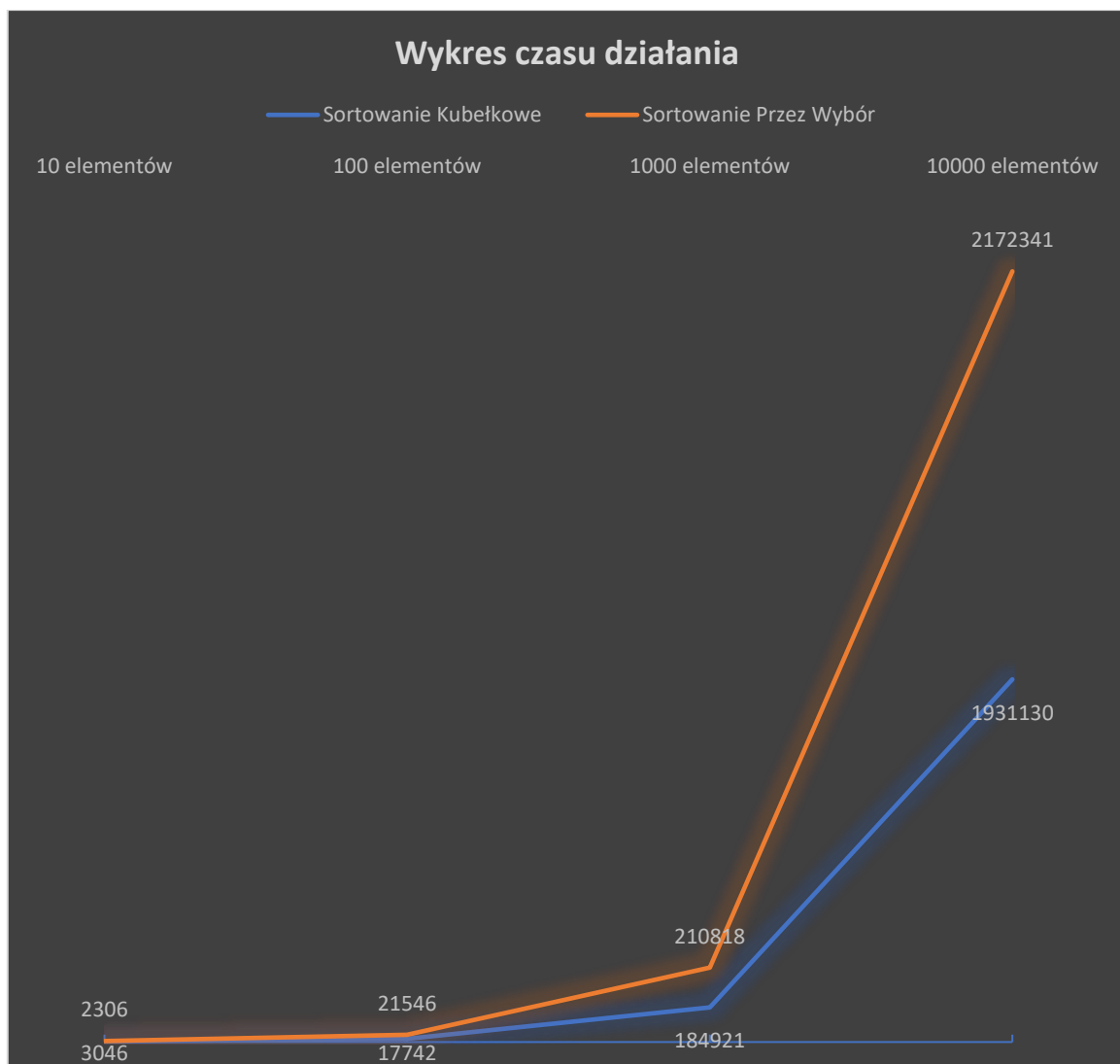
2 6 10 10 11 13 14 14 14 15 15 16 17 21 21 22 24 25 28 30

Czas wykonania algorytmu w mikrosekundach:

4375

Czas działanie algorytmów w mikrosekundach

	Sortowanie Kubełkowe	Sortowanie Przez Wybór
10 elementów	3046	2306
100 elementów	17742	21546
1000 elementów	184921	210818
10000 elementów	1931130	2172341
100000 elementów	19767932	24513331



Wnioski

Przy niedużych ilościach liczb nie ma znaczenia którego użyjemy ponieważ różnica czasowa jest niewielka lecz przy większych liczbach można zauważyć że sortowaniem szybszym jest sortowanie Kubełkowe ponieważ w tym sortowaniu nie musimy cały czas porównywać danych z poprzednią liczbą