ZADANIE 2

Celem ćwiczenia jest przeprowadzenie analizy złożoności obliczeniowej algorytmów. Ćwiczenie wykonywane jest na przykładzie wybranych algorytmów sortowania.

Badane algorytmy:

* algorytmy sortowania prostego:
  + przez wstawianie,
  + wybieranie,
  + zamianę (sortowanie bąbelkowe),
* algorytm sortowania przez scalanie,
* algorytm sortowania szybkiego.

Polecenia do wykonania:

1. **Przeprowadzić następującą analizę:**

Dla ciągu liczb całkowitych o długości podanego w danych do zadania (folder „Dane do polecenia 1”) wyznaczyć ("odręcznie" lub za pomocą programu) liczbę porównań oraz przesunięć (bądź zamian miejscami) elementów wykonaną przez poszczególne algorytmy sortowania prostego.

1. **Zapoznać się ze wskazaną pozycją literatury dotyczącą algorytmu sortowania szybkiego:** Niklaus Wirth – „Algorytmy + struktury danych = programy”, **rozdział 2.2.6**. Zwrócić uwagę na to, jak złożoność obliczeniowa algorytmu sortowania szybkiego zależy od wyboru elementu dzielącego x i od struktury sortowanego ciągu liczb. Odpowiedzieć na pytanie, jak złożoność obliczeniowa algorytmu sortowania szybkiego zależy od wyboru elementu dzielącego x i od struktury sortowanego ciągu liczb?
2. **Przeprowadzić następujący eksperyment obliczeniowy:**

Wyznaczyć czasy wykonywania obliczeń przez poszczególne algorytmy sortowania.

Obliczenia wykonać dla ciągów

* + - o różnych długościach =

1. 1000;
2. 10000;
3. 100000;
   * + o różnych strukturach, tzn. o elementach początkowo (przed sortowaniem):
4. uporządkowanych niemalejąco (od "najmniejszego do największego"),
5. ustawionych w przypadkowej (losowej) kolejności,
6. uporządkowanych nierosnąco ("od największego do najmniejszego").

Dla algorytmu sortowania szybkiego należy rozważyć następujące ustawienia ograniczenia x:

* + - (pierwszy lewy element podciągu),
    - (środkowy element podciągu).

**Omówić wyniki eksperymentu:**

* Porównać czasy wykonywania obliczeń dla stosowanych algorytmów.
* Opisać jak i dlaczego na czas wykonywania obliczeń przez poszczególne algorytmy wpływa:
* rozmiar rozwiązywanego problemu (długość ciągu);
* struktura danych – dane uporządkowane oraz ustawione w przypadkowej kolejności;
* Czy wszystkie otrzymane czasy wykonywania obliczeń są zgodne z przewidywaniami? Jeżeli nie, to wskazać które i spróbować wyjaśnić dlaczego.

**Uwagi:**

* Wartości elementów ciągu powinny być generowane z przedziału **;** przykład generowania liczb pseudolosowych z zadanego zakresu przedstawia program generowanie\_liczb\_losowych.cpp. (Uwaga C++: użycie srand() i rand() pozwala wygenerować wartości nie większe niż 32767).
* W sprawozdaniu należy umieścić zrzuty z ekranu pokazujące wartości 10 pierwszych elementów ciągów .