Sortowanie - Laboratorium nr 4 z PAMSI

Justyna Klijewska

23 03 2014

Zadanie do wykonania
W istniejącym już programie dodać sortowanie.

Program był pisany w środowisku Windows. Jest to wersja na ocenę 5. Zostały zrealizowane sortowania merge sort, quicksort i heap sort. Porównanie ich złożoności obliczeniowej znajduje się poniżej.

Sortowanie	Złożoność czasowa	Złożoność pamięciowa
Quicksort	O(nlogn)	Zależnie od implementacji
Mergesort	O(nlogn)	O(n)
Heapsort	O(nlogn)	O(n)

Tabela 1. Zależności między liczbą elementów w pliku a czasem wykonywania programu.

QUICKSORT

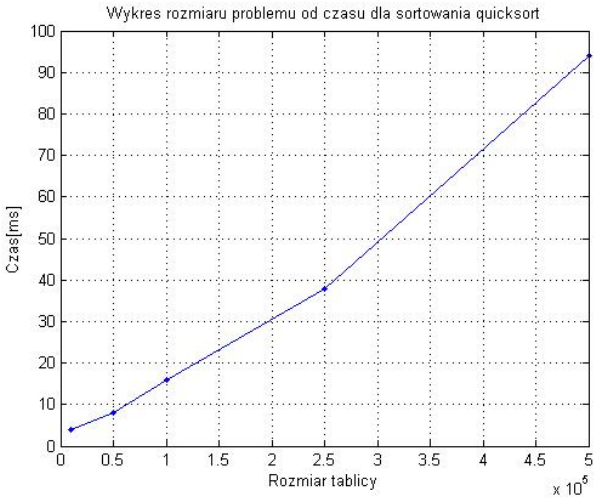
Jest to sortowanie szybkie i polega na zasadzie "dziel i zwyciężaj". Jego złożoność obliczeniowa to O(nlogn). Jest jednym z najczęściej używanych ze względu na szybkość wykonywania oraz łatwość implementacji.

MERGESORT

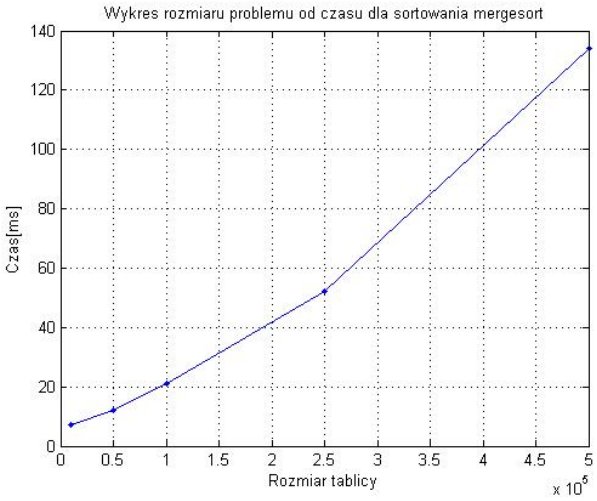
Jest to sortowanie przez scalanie i jego złożoność obliczeniowa to O(n log n). Podobnie jak quicksort korzysta z zasady "dziel i zwyciężaj".

HEAPSORT

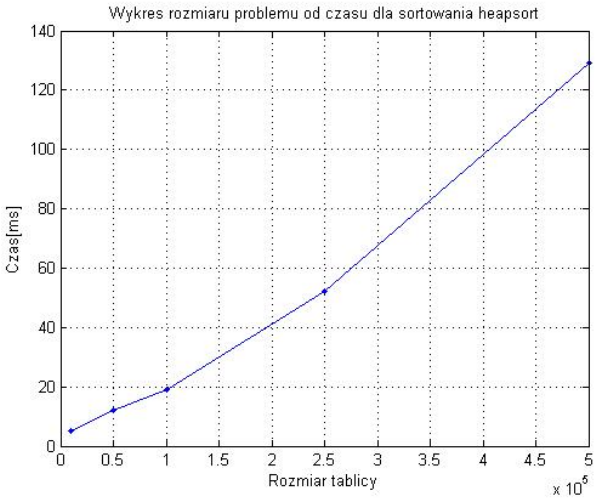
Jest to to sortowanie przez kopcowanie. Złożoność czasowa wynosi O(n log n), a pamięciowa – O(n). Jedną z największych jej zalet jest możliwość użycia tej samej tablicy w której znajdowały się nieposortowane elementy.



Rysunek 1: Wykres zależności liczby elementów w pliku od czasu wykonywania sortowania



Rysunek 2: Wykres zależności liczby elementów w pliku od czasu wykonywania sortowania



Rysunek 3: Wykres zależności liczby elementów w pliku od czasu wykonywania sortowania

WNIOSKI:

Najpopularniejszym i najczęściej używanym sortowaniem jest quicksort. I ciężko się dziwić, gdzyż w powyższych testach wykonywał on najszybciej swoje zadanie. Nie zmienia to faktu, że podane sortowania mają podobną złożoność obliczeniową, a co za tym idzie podobny czas wykonywania programu. Gdyby porównać powyższe z sortowaniem np.: bąbelkowym róznice byłyby większe.