Projektowanie algorytmów i metody sztucznej inteligencji

Przemysław Erbert 258964

wt 15:15

1 Link do repozytorium

https://github.com/Przemoerb/Pamsi2

2 Wprowadzenie

Zadanie projektowe polegało na przeszukaniu bazy danych "IMDb Largest Review Dataset" zawierajacych filmy oraz ich oceny, aby nastepnie gotowa już liste rankingowa posortować. W swoim programie do sortowania danych wykorzystałem 3 algorytmy: sortowanie przez scalanie, quicksort oraz sortowanie kubełkowe, a operacje te odbywaja sie na tablicy.

3 Opis poszczególnych funkcji

- Funkcja "fileLines" zwraca liczbe wierszy w pliku, który jest przekazywany jako argument funkcji. Funkcja ta otwiera plik, wczytuje linie za pomoca getline i zlicza ich ilość, a nastepnie zwraca te wartość.
- Funkcja "addFilm" ustawia wartości pól obiektu klasy Ranking na podstawie przekazanych argumentów. Id, średnia ocena i nazwa filmu sa przypisywane do odpowiednich pól w obiekcie.
- Funkcja "transferToArray" wczytuje dane z pliku "plik_ranking.tsv" i przenosi je do dynamicznie alokowanej tablicy obiektów klasy Ranking. Funkcja ta zaczyna od zadeklarowania tablicy o wielkości równa argumentowi amountOfDataToSort. Nastepnie otwiera plik, wczytuje jego pierwsza linie, a nastepnie dla każdej kolejnej linii (dopóki nie zostanie przeczytana ilość wierszy równa amountOfDataToSort) pobiera wartości id i średniej oceny oraz wywołuje funkcje searchMovieInformations w celu znalezienia nazwy filmu na podstawie id. Nazwa filmu oraz pozostałe dane sa przypisywane do obiektów w tablicy, a nastepnie tablica ta jest zwracana.

- Funkcja "transferToFile" zapisuje dane z tablicy obiektów klasy Ranking do pliku "plik_posortowany.txt". Funkcja ta zaczyna od otwarcia pliku, a nastepnie zapisuje wartości id, średniej oceny i nazwy filmu z każdego obiektu w tablicy do pliku. Funkcja ta kończy swoje działanie poprzez zamkniecie pliku.

- Funkcja "transferNamesToArray" wczytuje nazwy filmów z pliku "plik_nazwy.txt" i zapisuje je w dynamicznie alokowanej tablicy stringów. Funkcja ta zaczyna od zadeklarowania tablicy o wielkości równa ilości wierszy w pliku, a nastepnie otwiera plik, wczytuje jego pierwsza linie, a nastepnie dla każdej kolejnej linii zapisuje nazwe filmu w tablicy. Funkcja ta kończy swoje działanie poprzez zamkniecie pliku i zwrócenie tablicy z nazwami filmów.

- Funkcja "searchMovieInformations" przeszukuje tablice z nazwami filmów w celu znalezienia nazwy filmu na podstawie id. Funkcja ta działa na zasadzie wyszukiwania binarnego, gdzie szukane id jest porównywane z id filmu znajdujacego sie w środku tablicy. W zależności od wyniku porównania funkcja ta przechodzi do połowy tablicy zawierajacej poszukiwany element, aż do momentu znalezienia nazwy filmu lub wyjścia poza granice tablicy.

4 Opis sortowań

4.1 QuickSort

Mój kod implementuje algorytm sortowania quicksort dla tablicy struktur Ranking. Funkcja partition wybiera element pivot z tablicy i przeprowadza podział tablicy na dwie cześci: jedna zawierajaca elementy mniejsze od pivota i druga zawierajaca elementy wieksze od pivota. Funkcja quickSort rekurencyjnie wywołuje funkcje partition i sortuje obie cześci tablicy. Algorytm quicksort jest wydajnym algorytmem sortowania, który ma złożoność czasowa O(n log n).

4.2 BucketSort

Mój kod implementuje algorytm sortowania bucket sort dla tablicy struktur Ranking. Algorytm bucket sort polega na podziale danych na koszyki (buckets) i sortowaniu każdego z koszyków. W tym przypadku, zdefiniowano pieć koszyków, w których elementy zostaja przyporzadkowane na podstawie ich średniej oceny (avarageRating). Nastepnie każdy z koszyków zostaje posortowany za pomoca quicksort, a posortowane elementy zostaja ponownie połaczone w jedna posortowana tablice. Algorytm bucket sort ma złożoność czasowa O(n+k), gdzie k oznacza liczbe koszyków.

4.3 MergeSort

Ten kod implementuje algorytm sortowania przez scalanie (merge sort) dla tablicy obiektów typu Ranking. Funkcja merge() scala dwie posortowane podtablice w jedna posortowana tablice. Funkcja mergeSort() wykonuje rekurencyjnie sortowanie przez scalanie dla tablicy, dzielac ja na coraz mniejsze cześci, aż do pojedynczych elementów. Cały proces sortowania odbywa sie przez scalanie poszczególnych podtablic.

5 Wynik działania programu

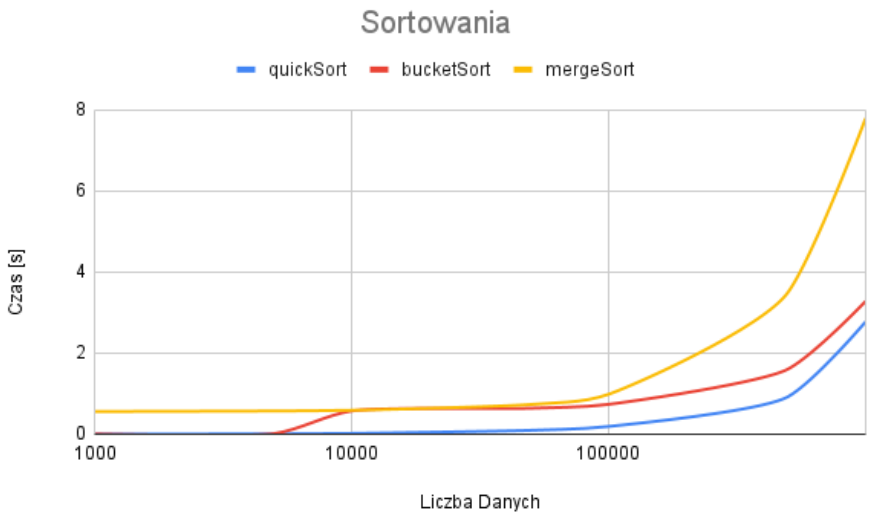

Figure 1: Szybkość sortowań

	Liczba Danych						
	1000	5000	10000	50000	100000	500000	1000000
QuickSort	0,001304	0,008332	0,019295	0,095094	0,189873	0,926937	2,77359
BucketSort	0,004282	0,013696	0,573092	0,639894	0,7361	1,60865	3,27957
MergeSort	0,558786	0,572616	0,589983	0,733625	0,984289	3,50718	7,79

Figure 2: tabela z czasem wykonywania poszczegolnych sortowań