Politechnika Wrocławska

Wydział Elektroniki, Fotoniki i Mikrosystemów

Projektowanie Algorytmów i Metody Sztucznej Inteligencji

Projekt 2

Zadanie na ocenę bdb

Prowadzący:
Dr inż. Łukasz Jeleń

Wykonała:
Zuzanna Mejer, 259382

Wrocław, 5 maja 2022r.

Spis treści

1. Wprowadzenie

Zadanie miało na celu zapoznanie się z algorytmami sortowania oraz przeprowadzenie analizy efektywności wybranych i zaimplementowanych sortowań. Z wymienionych algorytmów wybrałam sortowania: przez scalanie, szybkie oraz introspektywne.

2. Opis badanych algorytmów i ich złożoność obliczeniowa

2.1. Sortowanie przez scalanie

Jest to rekurencyjny algorytm sortowania danych, stosujący metodę „dziel i zwyciężaj". W algorytmie wyróżnia się trzy podstawowe kroki: podział danych wejściowych na 2 rozłączne podzbiory; rekurencyjnie zastosowanie sortowania dla każdego podzbioru, aż do uzyskania struktur jednoelementowych; scalenie posortowanych podzbiorów w jeden zbiór. Całkowita złożoność obliczeniowa dla sortowania przez scalanie wynosi $O(n \cdot logn)$, w związku z czym zastosowanie tego sortowania okaże się wydajniejsze dla bardzo dużych tablic.

2.2. Sortowanie szybkie

Również jest to algorytm sortowania danych stosujący metodę „dziel i zwyciężaj", nie wykorzystuje on jednak dodatkowych podtablic. Istnieje wiele implementacji sortowania szybkiego, jednak generalna idea jest taka, że wybierany jest jeden element w sortowanej strukturze, który nazywany jest piwotem. Może być to element środkowy, pierwszy, ostatni bądź losowy, przy czym należy pamiętać, że w przypadkach, kiedy piwot jest ciągle maksymalny lub minimalny, występuje najgorsza złożoność obliczeniowa $O(n^2)$. Przy optymalnych wyborach piwotu, złożoność wynosi $O(n \cdot logn)$.

2.3. Sortowanie introspektywne

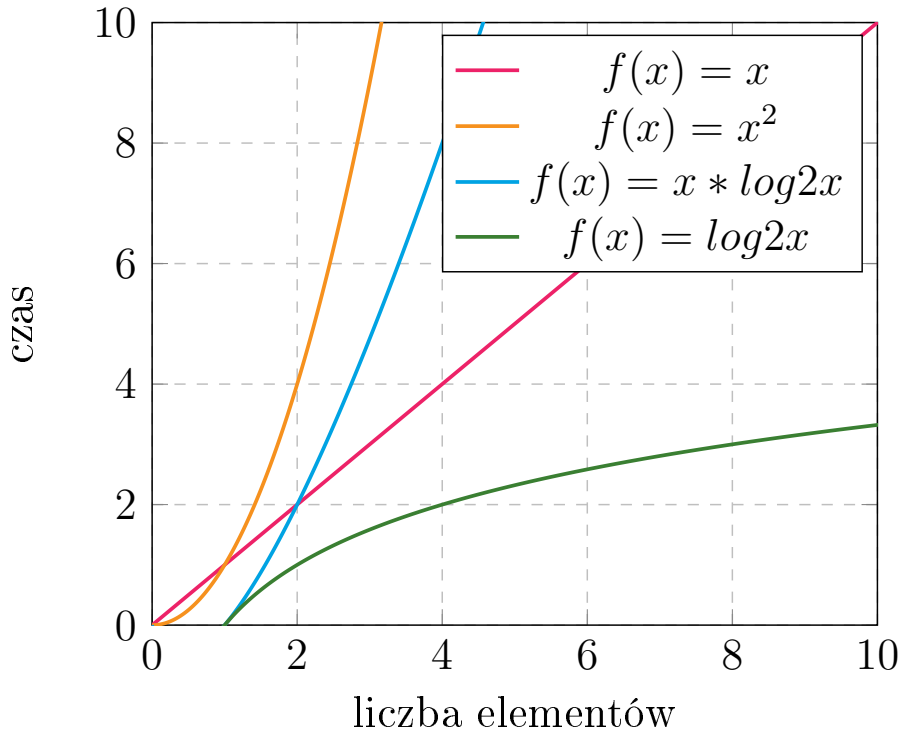
Jest to odmiana sortowania hybrydowego, które opiera się na spostrzeżeniu, że niewydajne jest wywoływanie ogromnej liczby rekurencji dla małych tablic w algorytmie sortowania szybkiego. Głównym założeniem algorytmu sortowania introspektywnego jest zatem wyeliminowanie problemu złożoności $O(n^2)$ występującej w najgorszym przypadku sortowania szybkiego. Sortowanie introspektywne jest połączeniem sortowania szybkiego i sortowania przez kopcowanie, które jest traktowane jako pomocnicze. Tym samym złożoność obliczeniowa wynosi $O(n \cdot logn)$.

2.4. Porównanie złożoności obliczeniowych wybranych algorytmów

Poniższa tabela zestawia oczekiwane i najgorsze przypadki złożoności wybranych algorytmów sortowania. Poniżej dodano także poglądowy wykres funkcji, na którym widać, że dla małej liczby danych sortowanie o złożoności kwadratowej będzie wydajniejsze niż dla logarytmicznej i przeciwnie dla dużej liczby elementów do posortowania.

Tab. 1: Porównanie oczekiwanych i najgorszych przypadków złożoności obliczeniowej dla wybranych algorytmów sortowania

	sortowanie		
	przez scalanie	szybkie	introspektywne
typowa złożoność	$O(nlong)$	$O(nlogn)$	$O(nlogn)$
najgorszy przypadek złożoności	$O(nlogn)$	$O(n^2)$	$O(nlogn)$



Rys. 1

3. Zadanie 1 - przeszukanie i przefiltrowanie danych

3.1. Krótki opis

Plik udostępniony do sortowania był okrojoną bazą filmów „IMDb Largest Review Dataset" ze strony kaggle.com. Plik zawierał tytuły filmów oraz przypisane im oceny. Niektóre pola z ocenami były puste, zatem przed wykonaniem zadań związanych z sortowaniem, należało wykonać przeszukanie i usunięcie wpisów bez ocen. Do wykonania tego zadania, zastosowano gotową strukturę z biblioteki STL: *std::vector*. Mimo chęci wykonania sortowań na strukturze dwuelementowej: *std::vector< std::pair <std::string, float> >*, przechowującej i tytuł filmu, i ocenę, komputery, na których wykonywałam testy złożoności obliczeniowej, nie były w stanie wykonać sortowań dla maksymalnej liczby elementów z pliku. Podsumowując, wykonane zostało przeszukiwanie, wykorzystujące strukturę jednoelementową, a następnie sortowane były jedynie oceny filmów.