Leonard Feszczuk 249027

gr. Pt 13:15-15:00

Projektowanie Algorytmów

i Metody Sztucznej Inteligencji

Projekt 1 - Algorytmy sortowania

1. **Wprowadzenie**

1.1 Celem projektu było zaimplementowanie i analiza efektywności trzech wy-branych algorytmów sortowania. Testy efektywności zostały przeprowadzone dla 100 tablic (typu całkowito-liczbowego) o następujących rozmiarach:

•10 000

•50 000

•100 000

•500 000

•1 000 000

Wykonano eksperymenty z sortowanie dla następujących przypadków:

•wszystkie elementy tablicy losowe

•25%, 50%, 75%, 95%, 99%, 99,7% procent początkowych elementów tablicy jest już posortowanych

•tablica posortowana jest malejąco.

**2 Badane algorytmy**

2.1 Sortowanie przez scalanie (mergesort)

Sortowana tablica dzielona jest rekurencyjnie na dwie podtablice aż do uzyskania tablic jednoelementowych. Następnie podtablice te są scalane w odpowiedni sposób, dający w rezultacie tablicę posortowaną. Wykorzystana jest tu metoda podziału problemu na mniejsze, łatwiejsze do rozwiązania zadania(„dziel i rządź”).

Złożoność obliczeniowa:

Średni przypadek: O(n)

Najlepszy przypadek: O(n)

Najgorszy przypadek: O(n)

2.2 Sortowanie szybkie (quicksort)

Na początku wybierany jest tzw. element osiowy. Następnie tablica dzielona jest na dwie podtablice. Pierwsza z nich zawiera elementy mniejsze od elementu osiowego, druga elementy większe lub równe, element osiowy znajdzie się między nimi. Proces dzielenia powtarzany jest aż do uzyskania tablic jednoelementowych, nie wymagających sortowania. Właściwe sortowanie jest tu jakby ukryte w procesie przygotowania do sortowania. Wybór elementu osiowego wpływa na równomierność podziału na podtablice (najprostszy wariant– wybór pierwszego elementu tablicy – nie sprawdza się w przypadku, gdy tablica jest już prawie uporządkowana).

Złożoność obliczeniowa:

Średni przypadek :O(nlogn)

Najlepszy przypadek: O(nlogn)

Najgorszy przypadek :O(n2)

2.3 Sortowanie introspektywne

Jest to metoda hybrydowa, będąca połączeniem sortowania szybkiego i sortowania przez kopcowanie. Sortowanie introspektywne pozwala uniknąć najgorszego przypadku dla sortowania szybkiego (nierównomierny podział tablicy w przypadku, gdy jako element osiowy zostanie wybrany element najmniejszy lub największy).Złożoność obliczeniowa:

Średni przypadek: O(nlogn)

Najlepszy przypadek: O(nlogn)

Najgorszy przypadek :O(nlogn)

**3 Wyniki**

3.1 Merge sort

3.2 Quick sort

3.3 Intro sort

**4. Porównanie elementów ze względu na rodzaj sortowania**

5. Wnioski

Zgodnie z założeniami najszybszym sortowaniem był intro sort a najwolniejszym merge sort chociaż w niektórych przypadkach był to quick sort, mogło się to wiązać z położeniem pivota w moim przypadku był on zawsze ustawiany na końcu. Defekt sortowania szybkiego niweluje sortowanie introspektywne jego hybrydowa natura pozwala nam uniknąć prawdopodobieństwa wybrania złego punktu odniesienia. Można jednak to prawdopodobieństwo dla sortowania szybkiego znacznie zmniejszyć wybierając losowy punkt odniesienia zamiast pierwszego lub ostatniego elementu

sortowanej tablicy. Posortowanie n pierwszych elementów tablic skraca zauważalnie czas sortowania dla każdego badanego algorytmu, to samo tyczy się tablic posortowanych malejąco. Powyższe wyniki potwierdzają teoretyczne założenia i poprawność zaimplementowanych algorytmów.

Źródła:

[1] Wikipedia

Merge sort

https://en.wikipedia.org/wiki/Merge

sort

[2] Wikipedia

Quick sort

https://en.wikipedia.org/wiki/Quicksort

[3] Wikipedia

Intro sort

https://en.wikipedia.org/wiki/Introsort

[4] Geeksforgeeks

Merge sort

https://www.geeksforgeeks.org/merge-sort/

[5] Geeksforgeeks

Intro sort

https://wwww.geeksforgeeks.org/know-your-sorting-algorithm-set-2-

introsort-cs-sorting-weapon