PAMSI PROJEKT 2

Łukasz Chwistek

14 czerwca 2021

# Opis projektu

W tym projekcie zostaje przeze mnie testowana efektywność algorytmów sortowania. Wszystkie programy zostały zrobione i przetestowane w programie MS Visual Studio 2019. Testowane algorytmy:

* 1. Sortowanie szybkie
  2. Sortowanie przez scalanie
  3. Sortowanie introspektywne

Algorytmy w projekcie były testowane dla różnych wielkości tablic:

1000, 100 000, 500 000, 1 000 000, maksymalna ilość danych z pliku.

Również zostały przeprowadzone testy gdy część tablicy jest już posortowana: 25%, 50%, 75%, 95%, 99%, 99,7% i dla odwrotnie posortowanej tablicy.

# Algorytmy

## Sortowanie przez scalanie (merge sort)

Rekurencyjny algorytm sortowania danych, stosujący metodę dziel i zwyciężaj. Szczególnie jest przydatny zwłaszcza przy danych dostępnych sekwencyjnie (po kolei, jeden element naraz). Asymptotyczny czas sortowania przez scalanie wynosi O(n log n)

## Sortowanie szybkie (quick sort)

Jeden z popularnych algorytmów sortowania działających na zasadzie „dziel i zwyciężaj”. Algorytm sortowania szybkiego jest wydajny: jego średnia złożoność obliczeniowa jest rzędu O(n log n). Ze względu na szybkość i prostotę implementacji jest powszechnie używany. Jego implementacje znajdują się w bibliotekach standardowych wielu środowisk programowania.

## Sortowanie introspektywne (introsort)

Odmiana sortowania hybrydowego, w której wyeliminowany został problem złożoności występującej w najgorszym przypadku algorytmu sortowania szybkiego.

Głównym założeniem algorytmu Sortowania Introspektywnego jest obsługa najgorszego przypadku algorytmu Sortowania Szybkiego tak, aby zapewnić logarytmiczno-liniową złożoność obliczeniową.

# Przebieg eksperymentów

## Elementy tablic losowe

## Elementy tablic posortowane w 25%

## Elementy tablic posortowane w 50%

## Elementy tablic posortowane w 75%

## Elementy tablic posortowane w 95%

## Elementy tablic posortowane w 99%

## Elementy tablic posortowane w 99,7%

## Elementy tablic posortowane w odwrotnej kolejności

# Wnioski

Na podstawie wykresów można dojść do wniosku, że najlepiej działa algorytm sortowania szybkiego. Jest on najszybszy oprócz swojego najgorszego przypadku, gdy jego złożoność wynosi . Jest to wykres dla 50% posortowania elementów. Dzieje się tak z powodu dobrania tzw. pivot’a w środku tablicy z liczbami. Jest to przyczyna nagłego wzrostu czasu potrzebnego do sortowania tablic. Sortowanie introspektywne jak i przez scalanie okazały się w miarę podobne pod względem szybkości. Sortowanie tablic częściowo posortowanych skraca czas działania algorytmów. Dla sortowania odwrotnych tablic sortowanie introspektywne okazało się strasznie słabe, dla 50 000 czas wynosi 653s. Może to być spowodowane nieodpowiednia implementacją algorytmu.

1. Bibliografia
2. https://pl.wikipedia.org/wiki/Sortowanie\_szybkie
3. https://pl.wikipedia.org/wiki/Sortowanie\_przez\_scalanie
4. https://pl.wikipedia.org/wiki/Sortowanie\_introspektywne