Badanie różnych metod sortowania

1. Badanie podstawowych metod sortowania

**Wnioski:**

Na podstawie danych metod z na każdym z wykresów widać, że zmiana rozmiaru tablicy powoduje zwiększenie czasu potrzebnego do wykonania sortowania liniowo, natomiast dynamika wzrostu jest różna dla Selection sorta, jest ona prawie wykładnicza zbliżając się do większych tablic, a dla Coctail sorta, wykres przypomina bardziej O(log n) w małym stopniu.

1. Badanie sortowania dla różnych kształtów tablic

**Wnioski**:

To na co należy zwrócić uwagę przy tych wykresach jest fakt, że dla każdego kształtu tablic Selection Sort wypada najgorzej, a Coctail Sort wraz z Insertion Sort porównywalnie najlepiej,

Natomiast Heap Sort w zależności od kształtu tablicy zachowuje się różnie tzn. dla Const jest bardzo szybki, lecz dla innych kształtów jego wyniki mocno się różnią.

1. Badanie Quicksort

* w dwóch wersjach: rekurencyjnie oraz iteracyjnie dla ciągu losowego

**Wnioski:**

Dla badanych algorytmów można zauważyć, że oba czasy wzrastają znacznie z zwiększeniem tablicy, jest to wzrost mniejszy od liniowego. W przypadku mojego komputera rekurencyjne rozwiązanie jest szybsze, niestety jest to związane z kompilatorem C#.

* Dla różnych kluczy(pivot) dla A-kształtnego ciągu

Badanie wykonane dla 3 różnych kluczy dla wersji z własnym stosem Quick Sorta.

public int RandPivot(int l, int p) // FUNKCJA DO LOSOWEGO KLUCZA

{

Random rnd = new Random();

int pivot = rnd.Next() % (p - l) + l;

return pivot;

}

public int DiffPivot(int l, int p, int i) // FUNKCJA WYBIERAJĄCA KLUCZ

{

switch (i)

{

case 0:

return (l + p) / 2;

case 1:

return p;

case 2:

return RandPivot(l, p);

default:

break;

}

return 0;

}

//x = t[(l + p) / 2];// analogicznie do wersji rekurencyjnej

//x = t[p];

//x = t[pivot = rnd.Next() % (p - l) + l];

x = t[DiffPivot(l, p, swi)]; // WYBIERANIE KLUCZA

Oprócz normalnej implementacji kodu, znajduję się w metodzie quick\_sort wybór klucz za pomocą wewnętrznej pętli podczas wywoływania badania, oraz switch które wybiarają klucz są one opisane powyżej.

**Wnioski:**

Jak widać najszybszy jest klucz oparty na medianie, ekstremalnie prawy klucz jest również ekstremalnie wolnym rozwiązaniem, a losowy klucz radzi sobie średnio 10 razy gorzej od klucza opartego na medianie.