Mikail Hüseyin AKAR

20060981

16.03.2022

Sıralama algoritmaları 1 ödev raporu

Sistem bilgileri:

Kullanılan dil: C

Kullanılan ortam: Visual Studio Code

Bilgisayar özellikleri:

OS: Windows 11 Home 21H2

CPU: I7-10750H

RAM: 16(2*8)GB DDR4 3200Mhz

Süreyi hesaplamak için kullanılan fonksiyon: <time.h> dosyasına ait clock() fonksiyonu

Kodlar:

```c
void selectionSort(int array[], int size){
  for (int i = 0; i < size - 1; i++){
    int min = i;
    for (int j = i + 1; j < size; j++){
      if (array[j] < array[min])
        min = j;
    }
    int temp = array[min];
    array[min] = array[i];
    array[i] = temp;
  }
}
```

Seçmeli sıralama (selection sort), her yinelemede sıralanmamış bir listeden en küçük öğeyi seçen ve bu öğeyi sıralanmamış listenin başına yerleştiren bir sıralama algoritmasıdır.

Zorluk: n^2

```c
void bubbleSort(int array[], int size){
  for (int i = 0; i < size - 1; ++i){
    for (int j = 0; j < size - i - 1; ++j){
      if (array[j] > array[j + 1]){
        int temp = array[j];
        array[j] = array[j + 1];
        array[j + 1] = temp;
      }
    }
  }
}
```

Kabarcık sıralama (bubble sort), iki bitişik öğeyi karşılaştıran ve istenen sırada olana kadar değiştiren bir sıralama algoritmasıdır.

Zorluk: n^2 (sıralı durumda: n)

```c
void insertionSort(int array[], int size){
  for (int i = 1; i < size; i++){
    int temp = array[i];
    int j = i - 1;
    while (temp < array[j] && j >= 0){
      array[j + 1] = array[j];
      --j;
    }
    array[j + 1] = temp;
  }
}
```

Eklemeli sıralama (insertion sort), sıralanmamış bir öğeyi her yinelemede uygun yerine yerleştiren bir sıralama algoritmasıdır.

Zorluk: n^2 (sıralı durumda: n)

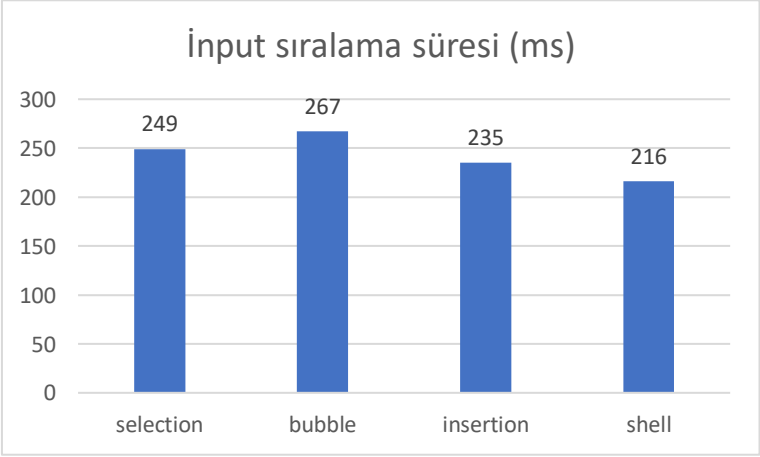
```c
void shellSort(int array[], int n){
  for (int interval = n / 2; interval > 0; interval /= 2){
    for (int i = interval; i < n; i += 1){
      int temp = array[i];
      int j;
      for (j = i; j >= interval && array[j - interval] > temp; j -= interval){
        array[j] = array[j - interval];
      }
      array[j] = temp;
    }
  }
}
```

Kabuk sıralama (shell sort), eklemeli sıralama algoritmasının genelleştirilmiş bir versiyonudur. Önce birbirinden uzaktaki öğeleri sıralar ve sıralanacak öğeler arasındaki aralığı art arda azaltır.

Zorluk: n*log(n) (tersten sıralı durumda: n^2)

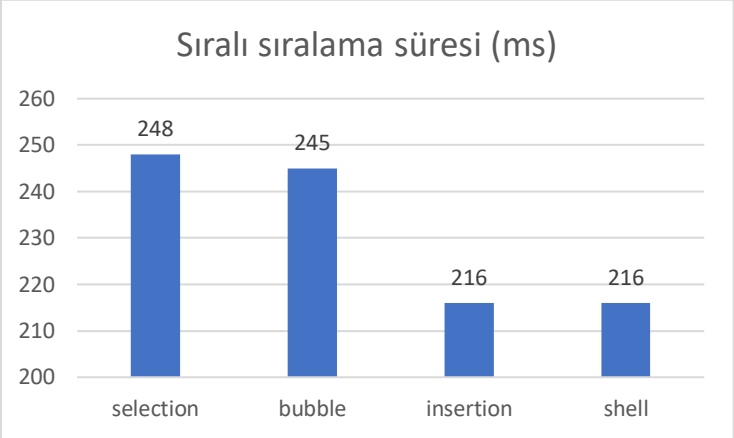
Sonuçlar:

1) input.txt için sıralama süreleri



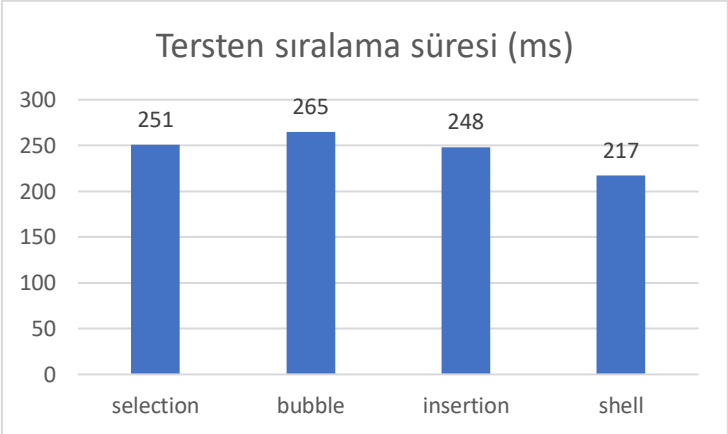
Shell sıralama algoritması input dosyasında yer alan sayıları küçükten büyüğe sıralamak için geri kalan sıralama algoritmalarına göre daha yüksek performans gösteriyor.

2) sirali.txt için sıralama süreleri



Dizinin küçükten büyüğe sıralı olması durumunda bubble ve insertion sıralama algoritmaları normalde gösterdiklerinden daha yüksek performans gösteriyorlar.

3) tersten_sirali.txt için sıralama süreleri



Dizinin tersten sıralı olması durumunda insertion sıralama algoritması çok daha düşük performans gösterirken selection ve shell algoritmaları hafif performans düşüşü gösteriyorlar.

Yorum:

Shell sıralama algoritması tersten sıralı dizilerde n^2 zorluluğa sahip olduğu için hafif bir performans düşüşü gösterse de, geri kalan durumlarda n*log(n) zorluluğa sahip olduğu için diğer algoritmalara göre daha yüksek performans gösteriyor.

Selection sıralama algoritması her durumda n^2 zorluluğa sahip olduğu için hepsinde çok yakın performanslar gösteriyor.

Bubble sıralama algoritması dizinin sıralı olması durumunda n zorluluğa sahip olduğundan geri kalan durumlardan (n^2 zorluluğa sahip olduğu durumlardan) daha yüksek performans gösteriyor.

Insertion sıralama algoritması n^2 zorluluğa sahip, tersten sıralı olması durumunda daha düşük performans gösterirken sıralı olması durumunda n zorluluğa sahip olduğu için daha yüksek performans gösteriyor.