

SIRALAMA ALGORİTMALARI

- Sıralama ve arama tekniklerinden pek çok programda yararlanılmaktadır. Günlük yaşamımızda elemanların sıralı tutulduğu listeler yaygın olarak kullanılmaktadır.
- Sıralama, sıralanacak elemanlar bellekte ise internal (içsel), kayıtların bazıları ikincil bellek ortamındaysa external (dışsal) sıralama olarak adlandırılır.

BUBBLE SORT

- Bir sonraki tarama ise bu en sağdaki eleman dışarıda bırakılarak gerçekleştirilmektedir. Bu dışarıda bırakma işlemi de dış döngüdeki sayaç değişkeninin değerinin her işletimde bir azaltılmasıyla sağlanmaktadır. Sayaç değişkeninin değeri 1 değerine ulaştığında ise dizinin solunda kalan son iki eleman da sıralanmakta ve sıralama işlemi tamamlanmaktadır.
- Bubble sort, sıralama teknikleri içinde anlaşılması ve programlanması kolay olmasına rağmen etkinliği en az olan algoritmalardandır (n elemanlı x dizisi için).

BUBBLE SORT

- Örnek:
- 9, 5, 8, 3, 1. rakamlarının azalan şekilde sıralanmasını kabarcık algoritmasıyla gerçekleştirelim.

BUBBLE SORT

- 1.Tur:

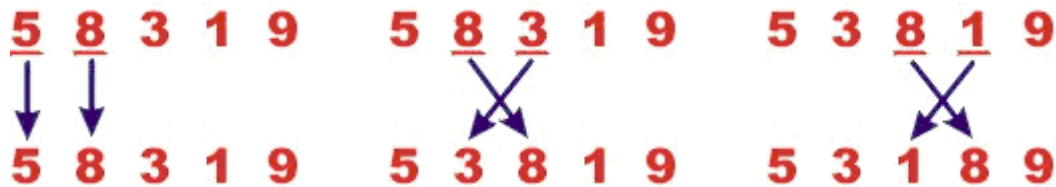


- 1. tur tamamlandığında en büyük eleman olan 9 en sona yerleşmiş olur ve bir daha karşılaştırmaya gerek yoktur.

SIRALAMA ALGORİTMALARI-BUBBLE SORT

- 2.Tur:

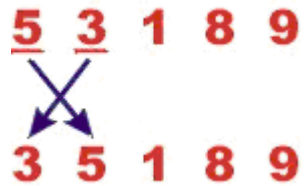
-



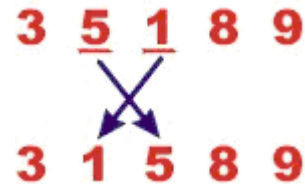
SIRALAMA ALGORİTMALARI-BUBBLE SORT

- 3.Tur:

5 3 1 8 9
3 5 1 8 9

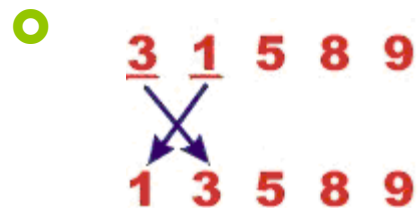


3 5 1 8 9
3 1 5 8 9



- 4.Tur:

- 3 1 5 8 9
1 3 5 8 9



BUBBLE SORT

```
○ public static void bubblesort(int [] x)
○ {   int n = x.Length;   int tut, j, gec;
○   for (gec=0; gec<n-1; gec++)
○       { for(j=0; j<n-gec-1; j++)
○           { if (x[j] > x[j+1])
○               {
○                   tut = x[j];
○                   x[j] = x[j+1];
○                   x[j+1] = tut;
○               }
○           }
○       }
○   }
```

○ en fazla (n-1) iterasyon gerektirir.

BUBBLE SORT

- Veriler : 25 57 48 37 12 92 86 33
- Tekrar 1 : 25 48 37 12 57 86 33 92
- Tekrar 2 : 25 37 12 48 57 33 86 92
- Tekrar 3 : 25 12 37 48 33 57 86 92
- Tekrar 4 : 12 25 37 33 48 57 86 92
- Tekrar 5 : 12 25 33 37 48 57 86 92
- Tekrar 6 : 12 25 33 37 48 57 86 92
- Tekrar 7 : 12 25 33 37 48 57 86 92

BUBBLE SORT

- Analizi kolaydır (İyileştirme yapılmamış algoritmada) :
- $(n-1)$ iterasyon ve her iterasyonda $(n-1)$ karşılaştırma.
- Toplam karşılaştırma sayısı : $(n-1)*(n-1) = n^2 - 2n + 1 = O(n^2)$
- (Yukarıdaki gibi iyileştirme yapılmış algoritmada etkinlik) :
- iterasyon i 'de, $(n-i)$ karşılaştırma yapılacaktır.
- Toplam karşılaştırma sayısı = $(n-1) + (n-2) + (n-3) + \dots + (n-k)$
$$= kn - k*(k+1)/2$$
$$= (2kn - k^2 - k)/2$$
- ortalama iterasyon sayısı, k , $O(k.n)$ olduğundan $= O(n^2)$

BUBBLE SORT

- Performans:
- Kabarcık sıralama algoritması ortalama $N^2/2$ karşılaştırma ve $N^2/2$ yer değiştirme işlemi gerçekleştirir ve bu işlem sayısı en kötü durumda da aynıdır.