Sıralama Algoritmaları

Yerleştirmeli Sıralama (Insertion Sort)

```
for ( i=1; i<n; i++) {
 for ( j=i; j>=0; j--)
  if (d[j] < D[j-1])
    yerdegis(j, j-1);
    else break;
}
```

Çalışma zamanı = n^2

Karşılaştırma Sayısı =n(n-1)/2

Direkt Yerleştirmeli Sıralama (Straight Insertion Sort)

```
for ( i=1; i<n; i++) {
 T=D[i];  j=i;
 while ( j > 0 && D[j-1] > T ) {
 D[j] = D[j-1];
 j--;
}
D[j] = T;
}
```

Çalışma zamanı = n^2

Karşılaştırma Sayısı =$n^2/4$

İkili Yerleştirmeli Sıralama (Binary Insertion Sort)

```
For ( i=1; i < n; i++) {
X=D[i];
Alt=0;
Ust=i-1;
while( Alt <= Ust) {
pivot = (Alt+Ust)/2;
if ( X < D[pivot] ) Ust = pivot – 1;
else Alt = pivot + 1;
}
for (j=i-1; j >= Alt; j--)
D[j+1] = D[j];
D[Alt]=X;
}
```

Çalışma zamanı = n log(n)
Karşılaştırma Sayısı =n (log(n) - log(e) ± 0,5)

Seçmeli Sıralama (Selection Sort)

```
for ( i=0; i < n; i++) {
min=i;
for ( j=i+1; j < n; j++)
if (D[j] < D[min])
min=j;
yerdegis(min, i);
}
```

Çalışma zamanı = n (n-1) /2
Karşılaştırma Sayısı =$n^2/2$

Kabarcık Sıralaması (Bubble Sort)

```
for ( i=n; i > 1; i--) {
for ( j=0; j < i-1; j++)
if (D[j] > D[j+1])
Yerdegis(j, j+1);
}
```

Çalışma zamanı = n^2
Karşılaştırma Sayısı =$n^2/2$

Hızlı Sıralama (Quick Sort)

```
quicksort(int Alt, int Ust) {
pivot=D[(Alt+Ust)/2];
Ustsinir=Ust;
Altsinir=Alt;
do {
  while ( pivot < D[Ustsinir])
  Ustsinir--;
  while ( pivot > D[Altsinir])
  Altsinir++;
  if ( Altsinir <= Ustsinir) {
  if ( Altsinir != Ustsinir)
  yerdegis(Ustsinir,Altsinir);
  Ustsinir--;
  Altsinir++;
 }
}while (Ustsinir >= Altsinir);
if ( Alt < Ustsinir)
quicksort( Alt, Ustsinir);
if ( Ust > Altsinir)
quicksort( Altsinir, Ust);
}
```

Çalışma zamanı = n log(n)
Karşılaştırma Sayısı = 2n ln(n)

Kümeleme kullanarak sıralama (Heap Sort)

```
downheap(int k, int n) {
T=D[k-1];
While ( k <= n/2) {
j=k+k;
if (( j<N ) && ( D[j-1] < D[j] )
j++;
if ( T >= D[j-1] )  break;
else { D[k-1] = D[j-1]; k=j; }
}
D[k-1] = T;
}

heapsort() {
for ( k = n/2; k > 0; k--)
downheap( k , n);
do {
yerdegis( 0 , n-1);
--n;
downheap( 1 , n);
} while ( n > 1);
}
```

Çalışma zamanı = n log(n)

Karşılaştırma Sayısı =2n log(n)

Kabuk Sıralaması (Shell Sort)

```
h=1;
while ( ( h*3+1) < n)
h=3*h+1;
while ( h > 0) {
for ( i = h-1; i < n; i++) {
B = D[i];
j = i;
for ( j = i; ( j >= h) && (D[j-h] > B ); j-=h)
 D[j] = D[j-h];
D[j] = B;
}
h /=3;
}
```

Çalışma zamanı = $n^{1,5}$
Karşılaştırma Sayısı =$n^{1,5}$