# Selection Sort

• Əsasən kiçikdən böyüyə sıralanır, lakin bu manipulasiya edilə bilir

•Big O Notationu $O(n^2)$ dır. Çünki bu alqoritmdə iç içə iki loop vardır.

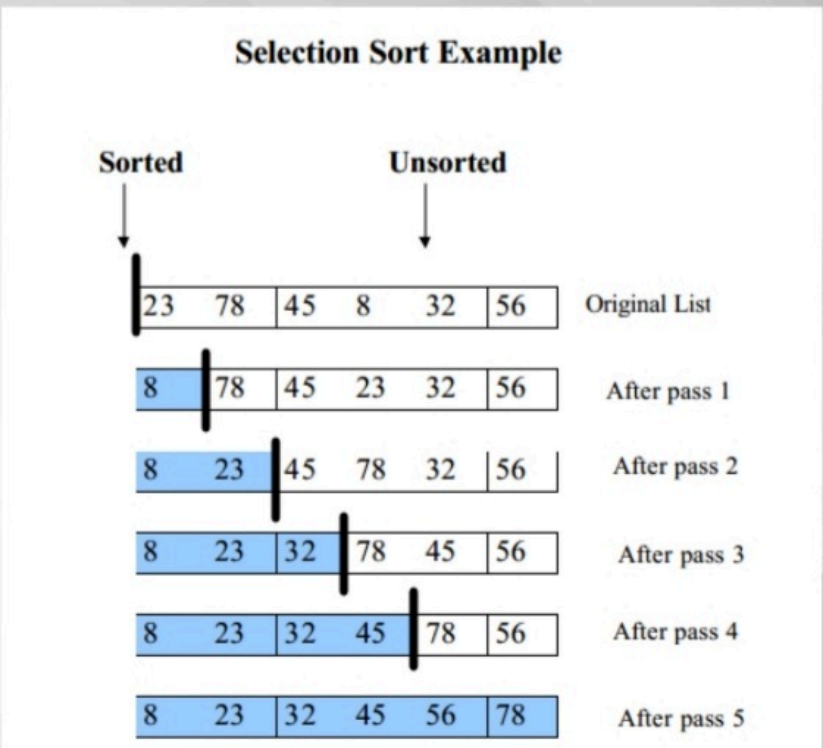
•Böyük arraylarda verimsizdir və daha pis performans göstərir.

•Yaxşı cəhəti odur ki O(n) swapından artıq swap etmir və bu yaddaşda daha az yer tutur

# Selection Sort

## QISA IŞLƏMƏ PRINSIPI

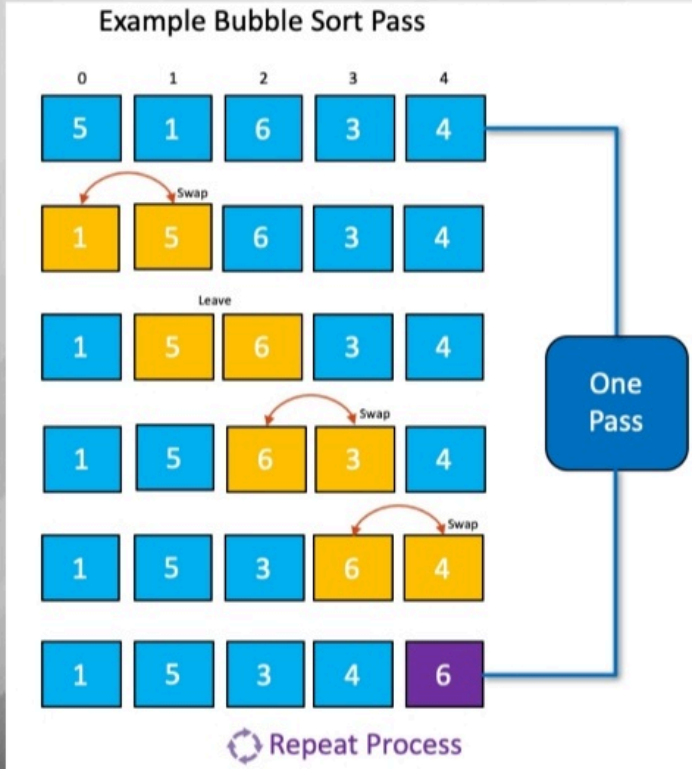
Göründüyü kimi əvvəlcə indexi 0 olan element ilə eyni arraydaki digər elementlər müqayisə edilir. Əgər hansısa bir element ilk elementdən kiçikdirsə swaplanır və ikinci addıma keçərkən indexi 1 olan element ilə dəvam olunur



# Bubble Sort

- Təməl məntiq- arraydaki ən böyük element arrayın sonuna atılır
- Kiçikdən böyüyə aparılır(manipulasiya edə bilərik)
- Linear olaraq qarşılaştırma edir(Yəni linear searchdaki kimi sıralı)
- Böyük saylar üçün uyğun deyil!
- Worst və Avarage Case $O(n^2)$, Best Case $O(n)$

# Bubble Sort




i indexi ilə alqoritmə başlanır. Eyni arraydaki i+1 indexi ilə müqayisə edilir. i>i+1 şərti ödənərsə swaplanır və bu proses dəvam edir. Bu yanlız bir iterationdur

# İnsertion Sort

- Sürüşdürmə ilə həyata keçrilir
- Prosed linear(sıralı) formada həyata keçir
- kiçikdən böyüyə sıralama
- Worst və Avarage Casedə Big O notation $O(n^2)$, Best casedə isə $O(n)$
- Böyük sayılarda sərfəsizdir

# İnsertion Sort

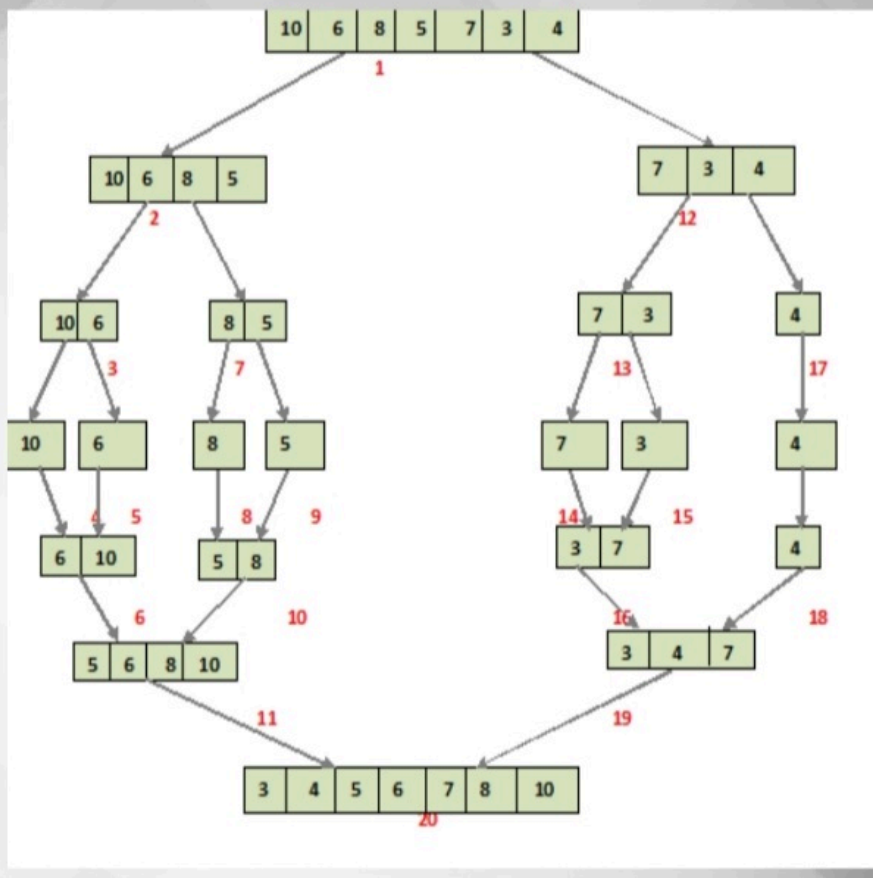
Müəyyən bir key verilir lakin bu key 0 dan yox, 1 ci indexdən başlayır. Sola baxılır və soldaki element böyükdürsə sağa sürüşdürülür. İlkinci addımda isə artıq Key=2 olur

# Merge Sort

- Parçala və fəthet məntiqində işləyir
- Big O notationu( 3 Casedə də ) O (log n ) dir.
- O log n olduğuna görə nəticəni daha tez verir.
- Çox elementi olan arraylərdə sərfəlidir.

# Merge Sort

Əvvəlcə massivi ən kiçik parçalarına çatana qədər ayırır. Tək tək ayrı arraylər formasına gətirditən sonra müqayisə edərək eyni addımları birləştirmə və sıralama vasitəsilə təkrarlayır



# Quick Sort

- Digərlərinə nisbətən olduqca tez bir alqoritmdir
- Best case də Big O notation O(log n)
- Çox elementi olan arraylarda çox sərfəlidir
- Əlavə yaddaş istifadə etmir
- Worst Case, pivot arraydaki ən kiçik vəya ən böyük element olaraq seçiləndə olur.
- Best Case pivot orta elementdə seçiləndə olur

# Quick Sort

Əvvəlcə bir pivot seçilir
Bizim edəcəyimiz alqoritm pivotdan böyük elementləri sağa,kiçikləri sola yerləştirəcəktir.
Tutaq ki i və j var. J nin göstərdiyi dəyər ilə pivot müqayisə edilir.
Element pivotdan böyükdürsə j++, kiçiksə i++ və i və j dəki elementlər swap edilir.
SON ADDIM:j  pivotdan bir əvvəlki elementə gəldiktə i++ və i və pivot swap. Artıq əvvəldə dedimizə çatdıq. İndi isə sağ və sol tərəfdə yeni pivot seçilərək alqoritm təkrarlanır