

Divide & Conquer

הפרד ומשול

- מחלקים את הבעיה למספר חלקים שווים (פחות או יותר).
- פועלים על כל חלק באופן ריקורסיבי
- מאחדים את התשובות הריקורסיביות.
- מטפלים במקרה הבסיס (עצירה).

מיון מיוזג – Merge Sort

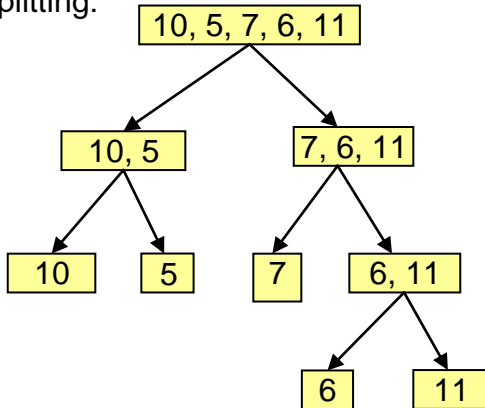
- אלגוריתם MergeSort זהו אלגוריתם קלאסי של שיטת הפרד ומשול.
- את מקרה הבסיס של מערך בגודל קטן (1), מטפלים באופן מיוחד (לא עושים כלום).
 - מחלקים את בעיית המיון לשני חלקים שווים (פחות או יותר).
 - ממיינים כל חלק באופן ריקורסיבי
 - את החלקים הממוינים מאחדים בזמן ליניארי.

```

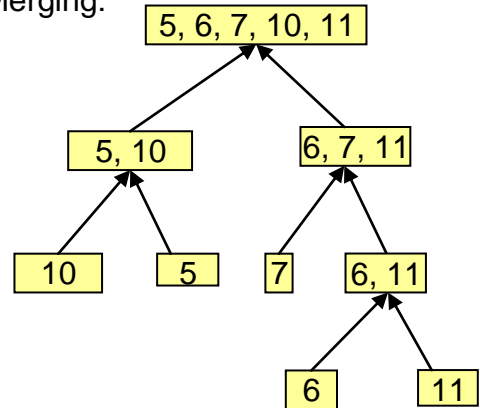
MergeSort (A)
  n ← length(A)
  if n > 1
    mid ← n/2
    A1 ← A [0 ... mid-1]
    A2 ← A [mid ... n-1]
    MergeSort(A1)
    MergeSort(A2)
    Merge(A1, A2, A)
  
```

דוגמא:

Splitting:



Merging:



פונקציית Merge() מקבלת שני מערכים A_1 , ו- A_2 ממוינים וממזגת אותם למערך ממוין יחיד A. מהו זמן ריצה הנדרש עבור Merge()?

```

Merge(A1, A2, A)
  ind1 = 0
  ind2 = 0
  for i ← 0 to i < length(A) ; i++
    if ind2 = length(A2)
      A[i] ← A1[ind1++]
    else if ind1 = length(A1)
      A[i] ← A2[ind2++]
    else
      if A1[ind1] < A2[ind2]
        A[i] ← A1[ind1++]
      else
        A[i] ← A2[ind2++]

```

זמן ריצה של Merge() הוא $O(n = n_1 + n_2)$

אנליזת זמן ריצה עבור MergeSort:

MergeSort(A)	
n ← length(A)	1
if n > 1	1
mid ← n/2	1
A1 ← A [0 ... mid-1]	n/2
A2 ← A [mid ... n-1]	n/2
MergeSort(A1)	T(n/2)
MergeSort(A2)	T(n/2)
Merge(A1, A2, A)	n

לכן את נוסחת חישוב זמן הריצה ניתן לרשום באופן רקורסיבי:

$$T(n) = \begin{cases} \Theta(1) & n = 1 \\ 2T(n/2) + \Theta(n) & n > 1 \end{cases}$$