



# خوارزمية الفرز بالدمج

المحاضرة الثالثة  
م. لمى السبع

# خوارزمية merge Sort

---

▪ **الهدف:** ترتيب قائمة  $a[1..n]$  ترتيباً تصاعدياً.

▪ آلة عمل الخوارزمية:

١. تقسم القائمة إلى قائمتين  $a[1,m]$  &  $a[m+1,n]$  وكل قائمة جزئية ترتب داخلياً.
٢. وأخيراً تدمج القائمتان الجزئيتان المرتبتان للحصول على قائمة مرتبة واحدة مرتبة.

❖ ملاحظة :

❖ تعتبر خوارزمية الفرز بالدمج أحد أنواع خوارزميات فرق-تجميع وهذا ما يوضحه المثال التالي.

divide

divide

divide

merge

merge

merge

## مثال توضيحي

➤ p: هي بداية المصفوفة ونبدأ بالدليل رقم

➤ R: دليل نهاية المصفوفة وفي المثال قيمتها ٧

➤  $Q = (p+r)/2$  هي دليل منتصف المصفوفة حيث

يتم تقسيم المصفوفة الى مصفوفتين جزئيتين .

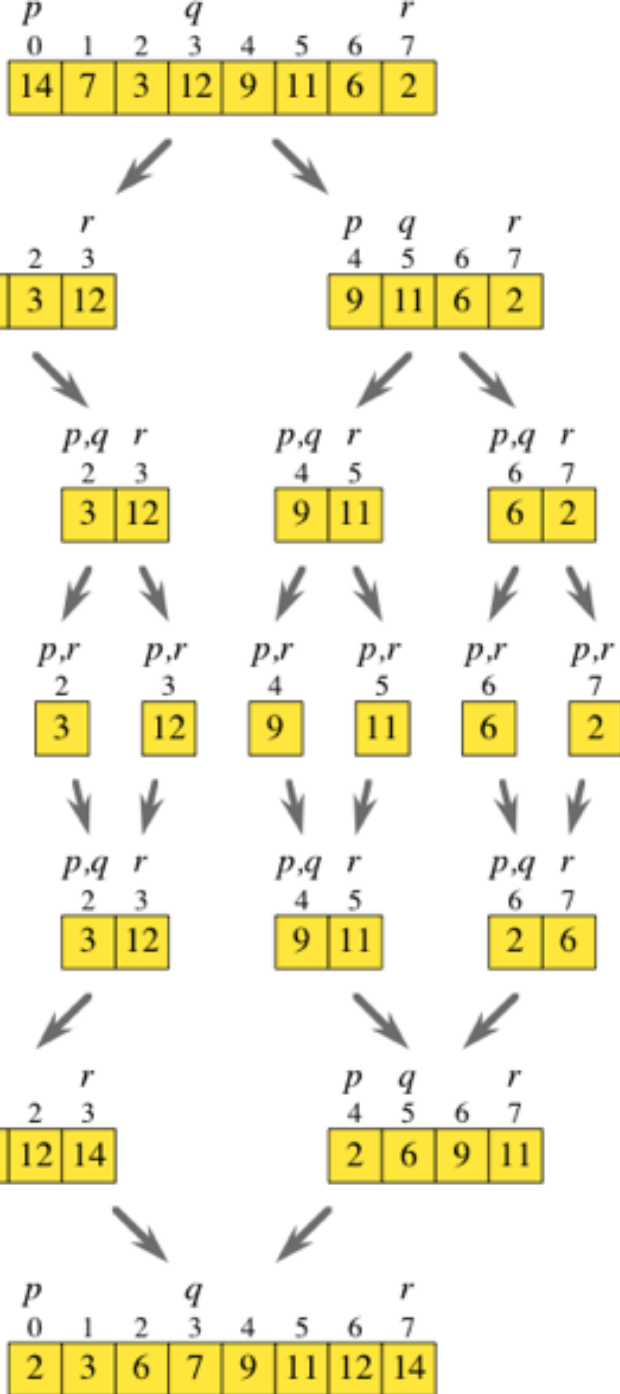
ونستمر بعملية التقسيم حتى تصبح قيمة  $p=r$  أي المصفوفة

تتكون من عنصر واحد فقط. وهذه هي المرحلة الأولى من

مراحل الخوارزمية وتدعى مرحلة فرق (تقسيم).

❖ وفي مرحلة التجميع يتم تشكيل مصفوفات مرتبة اعتماداً

على القيم الأحادية التي تم الوصول لها.



```
MergeSort(arr[], l, r)
```

```
If  $r > l$ 
```

1. Find the middle point to divide the array into two halves:

```
middle  $m = (l+r)/2$ 
```

2. Call mergeSort for first half:

```
Call mergeSort(arr, l, m)
```

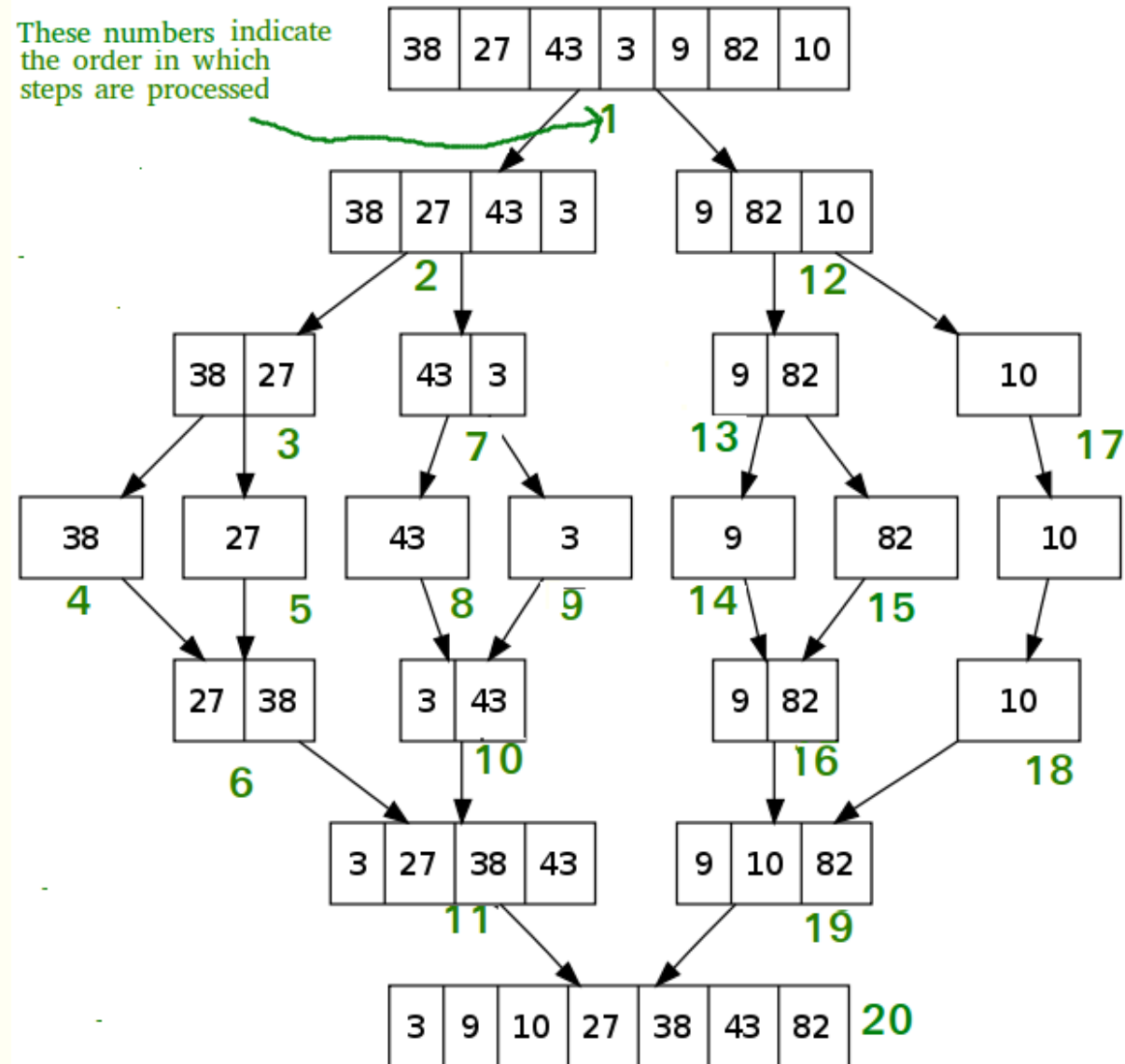
3. Call mergeSort for second half:

```
Call mergeSort(arr, m+1, r)
```

4. Merge the two halves sorted in step 2 and 3:

```
Call merge(arr, l, m, r)
```

## مثال آخر مع توضيح تسلسل العمليات



## الكود البرمجي لتابع mergeSort العودي

```
/* l is for left index and r is right index of the
   sub-array of arr to be sorted */
void mergeSort(int arr[], int l, int r)
{
    if (l < r)
    {
        // Same as (l+r)/2, but avoids overflow for
        // large l and h
        int m = l+(r-l)/2;

        // Sort first and second halves
        mergeSort(arr, l, m);
        mergeSort(arr, m+1, r);

        merge(arr, l, m, r);
    }
}
```

# الكود البرمجي لتابع merge

```
// Merges two subarrays of arr[].
// First subarray is arr[l..m]
// Second subarray is arr[m+1..r]
void merge(int arr[], int l, int m, int r)
{
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;

    /* create temp arrays */
    int L[n1], R[n2];

    /* Copy data to temp arrays L[] and R[] */
    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];

    /* Merge the temp arrays back into arr[l..r]*/
    i = 0; // Initial index of first subarray
    j = 0; // Initial index of second subarray
    k = l; // Initial index of merged subarray
    while (i < n1 && j < n2)
    {
        if (L[i] <= R[j])
        {
            arr[k] = L[i];
            i++;
        }
        else
        {
            arr[k] = R[j];
            j++;
        }
        k++;
    }

    /* Copy the remaining elements of L[], if there
    are any */
    while (i < n1)
    {
        arr[k] = L[i];
        i++;
        k++;
    }

    /* Copy the remaining elements of R[], if there
    are any */
    while (j < n2)
    {
        arr[k] = R[j];
        j++;
        k++;
    }
}
```

## وظيفة

---

### ■ المطلوب

١. اجراء بحث عن خوارزمية الفرز بالحشر (الاقحام)
٢. اجراء بحث عن خوارزمية الفرز بالاختيار