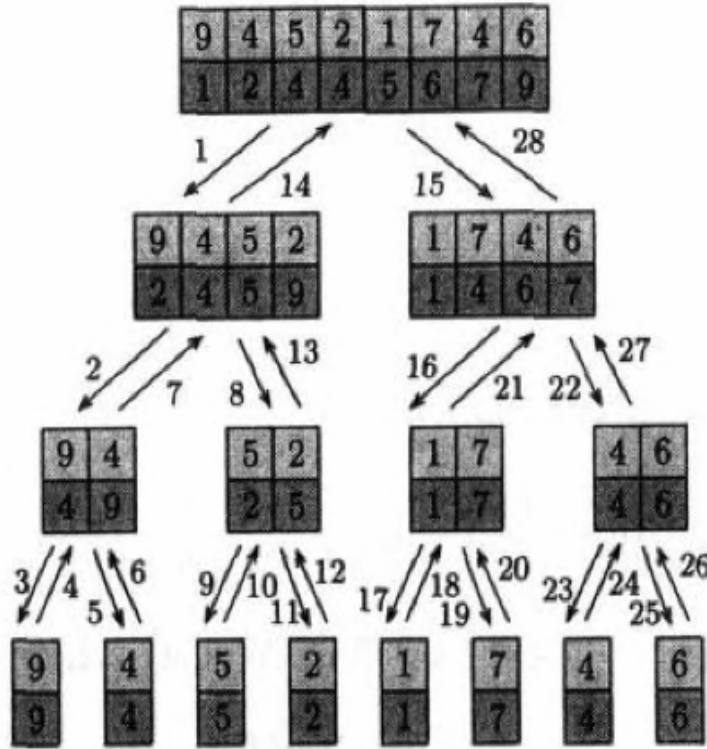


المحاضرة التاسعة

(Merge Sort) ترتيب الدمج

لترتيب قائمة $a[1..n]$ ترتيبا تصاعديا.
الفكرة/ تقسم القائمة $a[1..n]$ الى قائمتين $a[1:m]$ و $a[m+1:n]$. كل قائمة جزئية ترتب تداخليا. واخيرا تدمج القائمتان الجزئيتان المرتبتان للحصول على قائمة واحدة مرتبة. الشكل التالي يوضح سلوكية الخوارزمية على مجموعة من القيم:



الخوارزمية التالية تصف طريقة الترتيب هذه:

Algorithm MergeSort(A, low, high):

Input: An array A[1..n] of n elements.

Output: A[1..n] sorted in nondecreasing order.

1. **if** low < high **then**
2. mid $\leftarrow \lfloor (low + high) / 2 \rfloor$
3. MergeSort(A, low, mid)
4. MergeSort(A, mid+1, high)
5. Merge(A, low, mid, high)
6. **endif**

Algorithm Merge(A, p, q, r):

Input: An array A[1..m] of elements and three indices p, q and r, with $1 \leq p \leq q < r \leq m$, such that both the subarrays A[p..q] and A[q + 1..r] are sorted individually in nondecreasing order.

Output: A[p..r] contains the result of merging the two subarrays A[p..q] and A[q + 1..r].

1. **comment:** B[p..r] is an auxiliary array.
2. $s \leftarrow p$; $t \leftarrow q + 1$; $k \leftarrow p$
3. **while** $s \leq q$ and $t \leq r$
4. **if** $A[s] \leq A[t]$ **then**
5. $B[k] \leftarrow A[s]$
6. $s \leftarrow s + 1$
7. **else**
8. $B[k] \leftarrow A[t]$
9. $t \leftarrow t + 1$
10. **end if**
11. $k \leftarrow k + 1$
12. **end while**
13. **if** $s = q + 1$ **then** $B[k..r] \leftarrow A[t..r]$
14. **else** $B[k..r] \leftarrow A[s..q]$
15. **end if**
16. $A[p..r] \leftarrow B[p..r]$

تعقيدات الوقت
في الحالة الاسوأ:

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

$$\begin{aligned} T(n) &= 2T(n/2) + cn \\ &= 2[2T(n/2^2) + c(n/2)] + cn \\ &= 2^2 T(n/2^2) + 2cn \\ &= 2^2 [2T(n/2^3) + c(n/4)] + 2cn \\ &= 2^3 T(n/2^3) + 3cn \end{aligned}$$

⋮

⋮

⋮

$$= 2^k T(n/2^k) + kcn$$

بوضع $k = \log n$ ، $n = 2^k$

$$= nT(1) + cn \log n$$

$$= an + cn \log n$$

$$T(n) = O(n \log n)$$

تعقيب/ هذه الخوارزمية تفتنص طبيعة فرق- تسد لترتيب الدمج لكنها تعاني بعض الالكفاءه التي يمكن حذفها. حيث تتطلب خزن للمكدس يتناسب مع $\log n$. باستعمال ترتيب الحشر (insertion sort) عندما يصبح حجم القائمة صغيرا يقلص عمق التداخل ويحسن استعمال خزن المكدس والاحتساب.

```
if (high-low+1)<20 then
  Insertionsort(A,low,high)
else
  mid ← [(low + high)/2]
  MergeSort(A, low, mid)
  MergeSort(A, mid+1, high)
  Merge(A, low, mid, high)
endif
```

ملاحظة/ ادخال هذه التحسينات يبقي على تعقيدات الوقت $O(n \log n)$.

واجب/ وقت الحالة الاسوأ والحالة المتوسطة للخوارزمية MergeSort هو $O(n \log n)$. ما وقت الحالة الافضل؟ هل يمكن القول ان وقت الخوارزمية هو $\Theta(n \log n)$ ؟
واجب/ احسب تعقيدات الخزن للخوارزمية MergeSort.