

المحاضرة السادسة

البحث التعاقبي (Sequential search)

وهي طريقة للبحث عن عنصر معين x في مصفوفة من العناصر ولتكن A ، حيث توضح الخوارزمية التالية طريقة البحث هذه.

Algorithm SeqSearch(A, x):

Input: An array $A[1..n]$ of n elements and an element x .

Output: An index i such that $x=A[i]$, or the special value (0) if x dose not appear in A .

1. $i \leftarrow 1$
2. **while** ($i < n$) **and** ($x \neq A[i]$)
3. $i \leftarrow i+1$
4. **end while**
5. **if** $x=A[i]$ **then**
6. **return** i
7. **else**
8. **return** 0

تعقيدات الخزن:

$$S_{\text{SeqSearch}}(n) = \Theta(1)$$

تعقيدات الوقت:

أ- البحوث الناجحة:

(1) الحالة الافضل: تحدث عندما يكون عنصر البحث العنصر الاول في القائمة.

$$T_{\text{SeqSearch}}^B(n) = \Theta(1)$$

(2) الحالة الاسوأ: تحدث عندما يكون عنصر البحث العنصر الاخير في القائمة.

$$T_{\text{SeqSearch}}^W(n) = \Theta(n)$$

(3) الحالة المتوسطة:

$$T_{\text{SeqSearch}}^A(n) = \frac{\sum_{i=1}^n i}{n} = \frac{n+1}{2} = \Theta(n)$$

ب- البحوث الفاشلة:

$$T_{\text{SeqSearch}}(n) = \Theta(n)$$

ملاحظة/ توجد فكرة اخرى لتنفيذ طريقة البحث التعاقبي وكما موضح في الخوارزمية التالية:

Algorithm SeqSearch(A, x):

Input: An array $A[0..n]$ such that the n elements are stored in positions from 1 to n , and an element x .

Output: An index i such that $x=A[i]$, or the special value (0) if x dose not appear in A .

1. $i \leftarrow n$
2. $A[0] \leftarrow x$
3. **while** $A[i] \neq x$
4. $i \leftarrow i-1$
5. **end while**
6. **return** i

تحليل الخوارزمية: تعقيدات الوقت والخرن لهذه الخوارزمية هو نفسه للخوارزمية السابقة.

الترتيب بالحشر (Insertion sort)

وهي طريقة لترتيب مجموعة من العناصر ولتكن A، والخوارزمية التالية توضح ذلك.

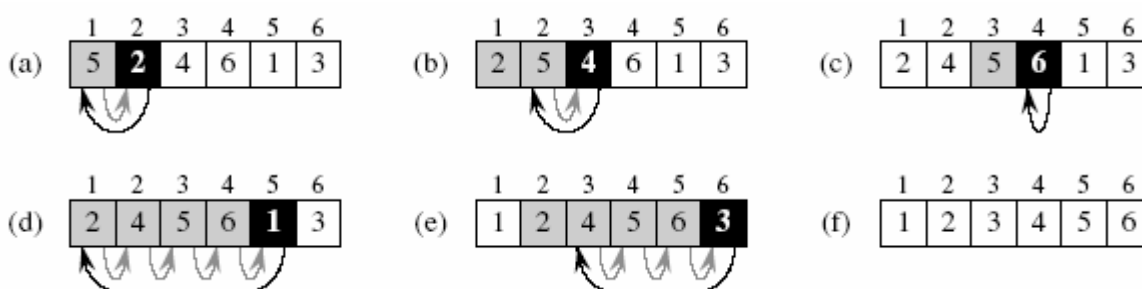
Algorithm InsertionSort (A) :

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

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

1. **for** i ← 2 **to** n
2. x ← A[i]
3. j ← i-1
4. **while** (j > 0) **and** (A[j] > x)
5. A[j+1] ← A[j]
6. j ← j-1
7. **end while**
8. A[j+1] ← x
9. **end for**

الشكل التالي يمثل مثالا توضيحيا على طريقة عمل هذه الخوارزمية:



تعقيدات الخرن:

$$S_{\text{InsertionSort}}(n) = \Theta(1)$$

تعقيدات الوقت:

الحالة الافضل: تحدث عندما تكون قائمة العناصر مرتبة ترتيبا تصاعديا.

$$T_{\text{InsertionSort}}^B(n) = \Theta(n)$$

الحالة الاسوأ: تحدث عندما تكون قائمة العناصر مرتبة ترتيبا تنازليا وجميع العناصر مختلفة.

$$T_{\text{InsertionSort}}^W(n) = \sum_{i=2}^n (i-1) = \sum_{i=1}^{n-1} i = \frac{n(n-1)}{2} = \Theta(n^2)$$

الحالة المتوسطة: في المتوسط نصف عدد العناصر في المصفوفة $A[1..i-1]$ هو اقل من $A[i]$ ونصفها الاخر اكبر، وعليه نحتاج تقريبا $(i/2)$ من المقارنات لحشر العنصر $A[i]$ في المصفوفة $A[1..i-1]$. لذلك فان العدد الكلي من المقارنات المنجزة من قبل الخوارزمية يوصف بالعلاقة التالية:

$$T_{\text{InsertionSort}}^A(n) = \sum_{i=2}^n (i/2) = \Theta(n^2)$$