المحاضرة الثالثة

مثال٥/ مسألة حساب المتوسطات السابقة (Prefix averages) لمجموعة من الاعداد.
مفهوم المسألة:- اذا كان لدينا مصفوفة معينة X مخصصة لخزن n من الاعداد الصحيحة، فالمطلوب في مسألة المتوسطات السابقة حساب مصفوفة اخرى ولتكن A بحيث ان العنصر A[i] يمثل متوسط قيم العناصر من X[1] الى X[i] لقيم i من 1 الى n وهذا يعني

$$A_i = \frac{\sum_{j=1}^{i} X_j}{i}$$

الخوارزمية التالية تقوم بحل هذه المسألة:

```
Algorithm  prefixAverages(X):
Input: An n-element array X of numbers.
Output: An n-element array of numbers A such that
        A[i] is the average of elements X[1],…,X[i].
1. for i← 1 to n
2.   a← 0
3.   for j← 1 to i
4.     a← a + X[j]
5.   end for
6.   A[i]← a /i
7. end for
8. return array A
```

يتصف مثال المسألة بـ n

تعقيدات الوقت باستخدام عد الخطوات:

1\. ……. n+1
2\. ……. n
3\. ……. $\sum_{i=1}^{n}(i+1)$
4\. ……. $\sum_{i=1}^{n} i$
6\. ……. n
8\. ……. n

$T_{prefixAverages}(n) = n^2 + 6n + 1$

واجب/ اعد صياغة الخوارزمية prefixAverages لتصبح تعقيداتها خطية وليست تربيعية.

الخوارزميات التداخلية (Recursive algorithms)

الخوارزميات التداخلية هي الخارزميات التي تستدعي نفسها حيث ان الطريقة الافضل لتحليل تعقيدات وقتها هو بكتابة علاقات تداخل (recurrence relations) تمثل تعقيدات الوقت لها، ثم بعد ذلك القيام بحل هذه العلاقات للحصول على تعقيدات الوقت.

مثال٦/ عودة للمسألة في المثال رقم ١.

تستخدم الخوارزمية التالية لحل هذه المسألة باستخدام اسلوب التداخل:

```
Algorithm Rsum(A,n):
Input: a positive integer n and an array A indexed
      from 1 to n.
Output:S,the sum of the numbers in A.
1. if n ≤ 0 then
2.  S ←  0
3. else
4.  S ←  Rsum(A,n-1)+ A[n]
5. end if
6. return S
```

يتصف مثال المسألة بـ n

تعقيدات الخزن: كل استدعاء يحتاج خليتان واحدة للمتغير S واخرى لعنوان العودة (return address) وبما ان عدد الاستدعاءات يحسب بالعلاقة التالية:

عمق التداخل= | الحجم الاولي في اول تنشيط – الحجم النهائي في اخر تنشيط | +١

= | 0 - n | +١

= n + ١

اذن تعقيدات الخزن هي

$$S_{Rsum}(n)= 2(n+1)$$

تعقيدات الوقت باستخدام اسلوب عد الخطوات: يصاغ الوقت على شكل علاقة تداخل ثم تستخدم طريقة التعويض التكراري (Iterative substitution) لحلها.

$$T_{Rsum}(n)=\begin{cases}3 & , \quad n\le 0\\ 3+T_{Rsum}(n-1) & , \quad n>0\end{cases}$$

$$\begin{aligned}T_{Rsum}(n) &= ٣ + T_{Rsum}(n-1)\\ &= ٣ + [٣ + T_{Rsum}(n-2)]\\ &= 2(٣) + T_{Rsum}(n-2)\\ &= 2(3) + [3 + T_{Rsum}(n-3)]\\ &= 3(3) + T_{Rsum}(n-3)\\ &\;\;.\\ &\;\;.\\ &\;\;.\\ &= m(3) + T_{Rsum}(n-m)\end{aligned}$$

عندما n=m يكون لدينا

$$T_{Rsum}(n)= 3n + 3$$

الحالات الافضل والاسوأ والمتوسطة (Best & worst & Average cases)

يمكن التخلص من الصعوبات في الحالات التي تكون فيها المعاملات المختارة(خصائص المثال) غير مناسبة وحدها لتحديد عدد الخطوات(العمليات) من خلال تعريف ثلاث انواع من عد الخطوات (العمليات):-

١) عد خطوات(عمليات) الحالة الافضل:- وهو ادنى عدد من الخطوات(العمليات) يمكن انجازها لمعاملات معينة.

٢) عد خطوات(عمليات) الحالة الاسوأ:- وهو اقصى عدد من الخطوات(العمليات) يمكن انجازها لمعاملات معينة.

٣) عد خطوات(عمليات) الحالة المتوسطة:- وهوالعدد المتوسط من الخطوات(العمليات) التي يمكن انجازها على امثلة مسألة بمعاملات معينة.

مثال٧/ ايجاد العنصرالاكبر.

```
Algorithm  arrayMax(A,n):
Input: An array A storing n integers.
Output: currentMax,the maximum element in A.
    currentMax←  A[1]
    for i← 2  to  n
      if currentMax < A[i] then
       currentMax←  A[i]
      end if
    end for
    return currentMax
```

يتصف مثال المسألة بـ n

تعقيدات الوقت باستخدام اسلوب عد الخطوات:

الحالة الافضل : تحدث عندما A[1] هو العنصر الاكبر

$$T_{arrayMax}^{B}(n) = 2n + 1$$

الحالة الاسوأ : تحدث عندما تكون A مرتبة تصاعديا

$$T_{arrayMax}^{W}(n) = 3n$$

الحالة المتوسطة: تحسب متوسط التعقيدات لجميع الحالات الممكنة لقيم المصفوفة

$$T_{arrayMax}^{A}(n) = \frac{\sum_{i=1}^{n}(2n+i)}{n} = \frac{5n+1}{2}$$