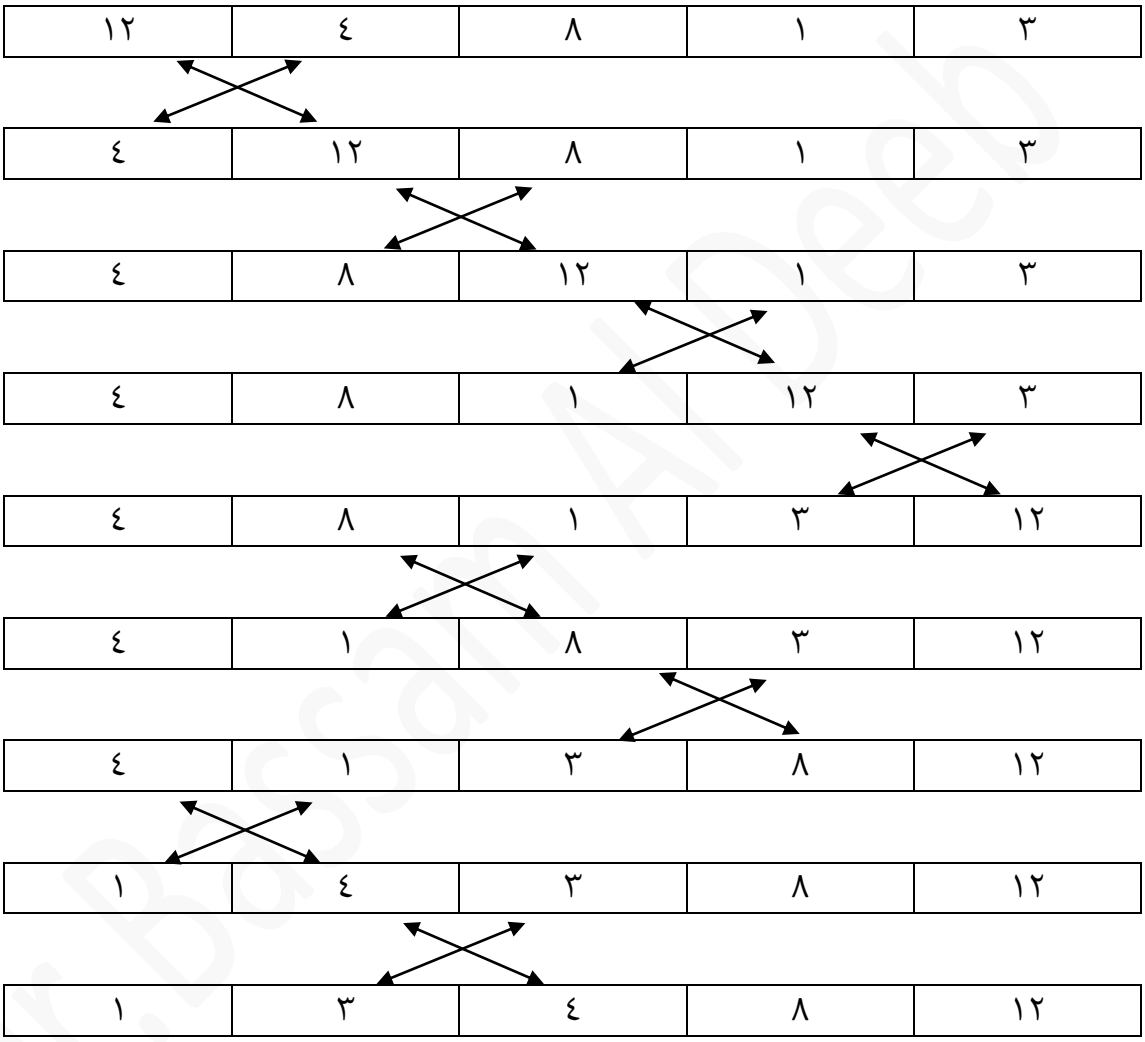
خوارزمية الفرز الفقاعي Bubble Sort

تعمل هذه الخوارزمية على مبدأ الفقاعات الكبيرة تطفو و الصغيرة ترسو ويتم ذلك من خلال المرور على عناصر المصفوفة أكثر من مرة . في كل مرة تتم المقارنة بين كل عنصرين متتاليين ومبادلتهما . عندما يكون العنصر الثاني هو الأصغر عند الترتيب التصاعدي وبالعكس عند الترتيب التنازلي

مثال:



خطوات عمل الخوارزمية :

١. المرور على عناصر المصفوفة من خلال حلقتين متداخلتين
٢. مقارنة كل عنصرين متتاليين ومبادلتهم اذا كانا غير مرتبين
٣. عند الانتهاء من المرور الأول نكون قد انجزنا n-1 مقارنة وازاحة العنصر الأكبر الى الخانة الأخيرة
٤. نكرر المرور السابق بعد تجاهل العناصر المرتبة في كل مرة بحيث يتم الترتيب.

شيفرة الخوارزمية :

```
Bubblesort(int a []) {
  int last = a.length-1;
   for(int i=0 ; i<a.length ;i++)
      for(int j=0 ; j<last-i ;j++)
         If(a[j] >a[j+1])
          Swap(a,j,j+1)
}
Swap(int a[] , int e1 , int e2){
int temp;
temp = a [e1];
a[e1] = a[e2];
a[e2] = temp;
}
```

مثال تطبيقي :

ليكن لدينا المصفوفة a المعرفة بالشكل :

5	1	8	7	2
0	1	2	3	4

ونريد ترتيبها وفق خوارزمية الفرز الفقاعي .

في البداية نمرر المصفوفة في الدالة Bubblesort ويتم ضمنها تمهيد المتغير last=4

ومن ثم الدخول في الحلقة الأولى i=0

ومن ثم الحلقة الثانية j=4-0=4 وحتى j=0

وتتم المقارنة بين كل عنصرين متتاليين وفي حال كانا غير مرتبين

يتم استدعاء الدالة swap لمبادلتهما ونلاحظ ضمن الحلقة الثانية تتم المبادلات

J=0 1 5 8 7 2

J=1 x

J=2 1 5 7 8 2

J=3 1 5 7 2 8

J=4 x

ثم يعود الى الحلقة الأولى "الخارجية" لزيادة العداد i=1

والدخول في الحلقة الثانية j=3 ⟵ j=0

<u>تجاهل العنصر الأخير</u>

J=0 x لا يوجد مبادلة

J=1 x

J=2 1 5 2 7 8

J=3 x

ثم الخروج الى الحلقة الخارجية لزيادة العداد من جديد i=2

j=2 ⟵ j=0

J=0 x

J=1 1 2 5 7 8

J=2 x

ثم الخروج لزيادة عداد i i=3

ونلاحظ ان المصفوفة مرتبة وبالتالي يخرج من الحلقة الثانية دون خطوات منفذه لزيادة العداد i=4

واقفا دون خطوات منفذه ويعيد المصفوفة مرتبة .

Dr Bassam Al Deeb

خوارزمية الفرز بالحشر insertion sort:

فكرة هذه الخوارزمية هي مقارنة كل عنصر مع جميع العناصر التي تسبقه وتبدل مكتنه ان وجد عنصر اكبر منه .

حتى تصبح العناصرالسابقة له جميعها اصغر منه .

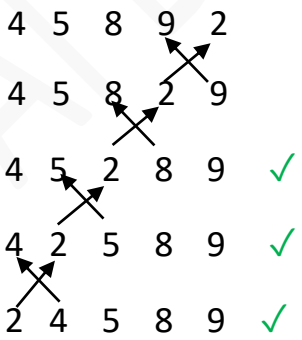
مثال: على فرض اننا نريد ترتيب المصفوفة 5 , 8 , 4 , 9 , 2

نبدأ من العنصر الثاني 5 8 4 9 2 x

ثم العنصر الثالث 5 4 8 9 2 ✓

4 5 8 9 2 ✓

ثم العنصر الرابع



وبذلك تكون المصفوفة قد اصفحت مرتبة

خوارزمية الفرز بالحشر insertion sort:

```
insertionsort(int a[]) {
int temp;
    for(int i=1 ; i< a.length ; i++)
        for(int j=i ;  j< 0 ; j - -)
            if( a[j] <  a[j-1] )}
             temp = a[j];
            a[j]=a[j-1];
            a[j-1] = temp ;} }
```

شرح عمل الشيفرة :

١. تم تعريف اجراء insertion sort وتمرير له المصفوفة a
٢. تعريف متغير وسيط temp لاستخدامه في المبادلة
٣. الدخول في حلقة أولى تمر على العناصر بدءاً من العنصر الثاني i=1
٤. الدخول في حلقة ثانية بدءاً من العنصر الذي وصلت إليه الحلقة الأولى j=i وهي حلقة عكسية متناقصة تستمر حتى " العنصر الثاني "
٥. مقارنة كل عنصر مع العنصر السابق له داخل الحلقة الثانية والمبادلة عند الضرورة وتستمر حتى بداية المصفوفة
٦. في نهاية الحلقة الداخلية يعود التحكم الى الحلقة الخارجية لزيادة قيمة العداد i والعودة من جديد حتى يتم الترتيب

مثال تطبيقي :

2	1	6	3	4	5	9
0	1	2	3	4	5	6

i=1 j=1

a[1] < a[0] swap

1	2	6	3	4	5	9
0	1	2	3	4	5	6

i=2

j=2

a[1] < a[0] not swap

j=1 a[j] < a[0] not swap

i=3 j=3

a[3] < a[2] swap

1	2	3	6	4	5	9
0	1	2	3	4	5	6

J=2 a[2] < a[1] not swap

J=1 a[1] < a[0] not swap

i=4 j=4

a[4] < a[3] swap

1	2	3	4	6	5	9
0	1	2	3	4	5	6

j=3 a[3] < a[2] not swap

j=2 a[2] < a[1] not swap

j=1 a[1] < a[0] not swap

i=5 j=5

a[5] < a[4] swap

1	2	3	4	5	6	9
0	1	2	3	4	5	6

j=4 a[4] < a[3] not swap

j=3 a[3] < a[2] not swap

j=2 a[2] < a[1] not swap

j=1 a[1] < a[0] not swap

i=6 j=6

j=6 a[6] < a[5] not swap

j=5 a[5] < a[4] not swap

j=4 a[4] < a[3] not swap

j=3 a[3] < a[2] not swap

j=2 a[2] < a[1] not swap

j=1 a[1] < a[0] not swap

وتكون المصفوفة مرتبة .

Dr.Bassam Al Deeb

خوارزمية Shell

تعمل خوارزمية shell على تطوير خوارزمية الحشر بهدف الاختصار من زمن التنفيذ وذلك من خلال تقسيم العمل الى مراحل .

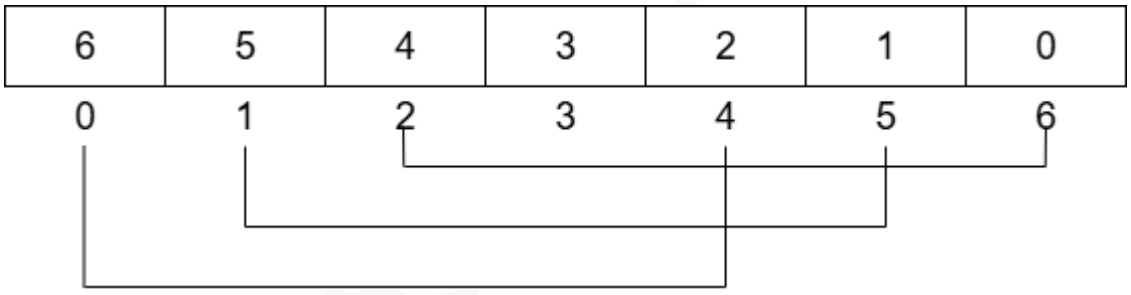
طريقة عمل الخوارزمية: تعمل الخوارزمية على مراحل وفق المثال التالي :

6	5	4	3	2	1	0
0	1	2	3	4	5	6

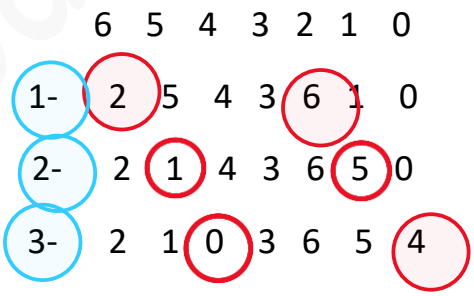
المرحلة الأولى : تأخذ الخوارزمية مسافة بين العناصر المطلوب ترتيبها مقدارها k=n/2 حيث n حجم المصفوفة

n=7 ⟶ k=7/2 ≈ 4

- نقارن العنصر a[4] مع العنصر الذي يبعد عنه مسافة k=4 أي العنصر a[0] وتتم المبادلة عند اللزوم.
- الانتقال في العنصر التالي a[5] ومقارنته بالعنصر الذي يبعد عنه المسافة k أي مع a[1] والمبادلة عند اللزوم .
- وهكذا تستمر حتى حتى العنصر الأخير ونكون قد انهينا المرحلة الأولى



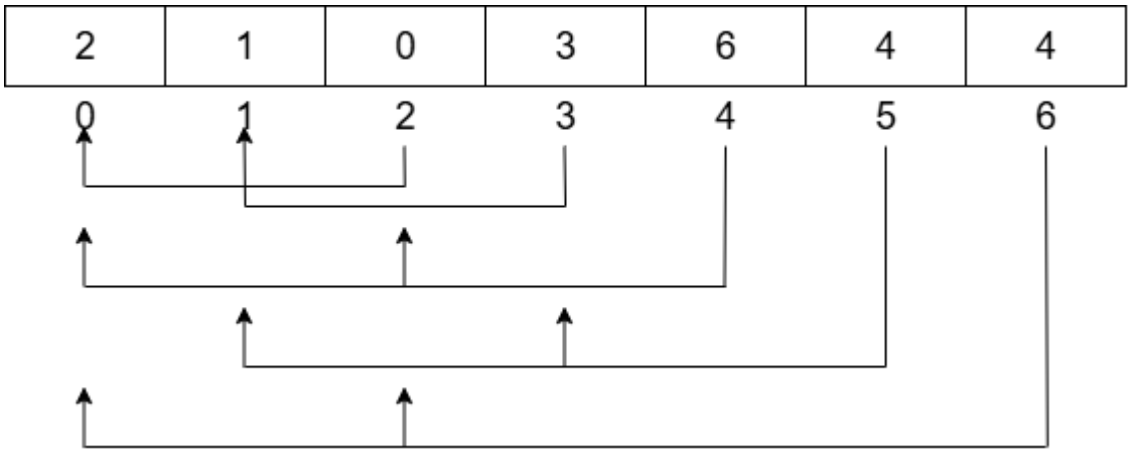
وتكون النتيجة بعد اجراء المبادلات اللازمة في هذه المرحلة .



المرحلة الثانية : في هذه المرحلة نحسب قيمة k الجديدة

k=n/2 k=4/2=2

وبالتالي المسافة بين العناصر التي تم مقارنتها هي ٢ ونعيد خطوات المرحلة الأولى على المسافة الجديدة .



1- 0 1 2 3 6 5 4

2- not swap

3- not swap

4- not swap

5- 0 1 2 3 4 5 6

بالتالي تكون قد ترتبت المصفوفة .

الكود البرمجي لخوارزمية Shell :

```
shellsort(int a[]) {
int n = a.length;
for(int k=n/2 ; k>1 ; k=k/2)
insertionsort(a,k); }
insertionsort(int a[] , int k){
int temp;
for(int i=k ; i< a.length ; i ++)
for(int j=i ; j>=k ; j=j-k)
if(a[j] < a[j-k]){
        temp = a[j];
         a[j]=a[j-k];
    a[j-k] = temp; }}
```

شرح الشيفرة :

١. تعريف إجرائية shellsort وتمرير المصفوفة لها
٢. تعريف المتغير n يدل على طول المصفوفة
٣. الدخول في الحلقة بدءاً من القيمة n=k/2 منتصف المصفوفة وتستمر الحلقة طالما k>1 (هذه الحلقة تقفز خطوات بمقدار k/2 في كل مرة .
٤. استخدام الإجرائية insertionsort وتمرير المصفوفة a مع قيمة k في الحلقة حيث تعمل هذه الإجرائية على مبادلة العنصر ذو الفهرس k مع العناصر السابقة له عند الضرورة بخطوة k كل مرة.
٥. الإجرائية insertionsort :
   - ❖ تعريف متغير وسيط
   - ❖ الدخول في حلقة أولى بدءاً من k وحتى اخر المصفوفة
   - ❖ الدخول في حلقة ثانية بدءاً i وحتى k بخطوات مقدارها k تناقصيا كل مرة
   - ❖ اجراء المقارنة بين كل عنصرين بينهما مسافة k والمبادلة عند اللزوم .

مثال تطبيقي:

6	5	4	3	2	1	0
0	1	2	3	4	5	6

n=7 k=7/2=4;
insertionsort(a ,4)

i=4 j=4 ;

a[4] < a[0] swap

2	5	4	3	6	1	0
0	1	2	3	4	5	6

j=0 not swap

j=5 a[5] < a[1] swap

2	1	4	3	6	5	0
0	1	2	3	4	5	6

j=5-4=1 not swap

k=4/2=2

insertionsort(a ,2)

i=2 j=2 a[2] < a[0] swap

0	1	2	3	6	5	4
0	1	2	3	4	5	6

j=0 not swap

i=3 j=3

a[3] < a[1] not swap

j=1 ; not swap

i=4 j=4 a[4] < a[2] not swap

j=2 a[2] < a[0] not swap

j=0 not swap

i=5

j=5 a[5] < a[3] not swap

j=3 a[3] < a[1] not swap

j=1 not swap

i=6 j=6 a[6] < a[4]

0	1	2	3	4	5	6
0	1	2	3	4	5	6

j=4 not swap

j=2 not swap

j=0 not swap

وتكون المصفوفة قد أصبحت مرتبة.

Dr.Bassam Al Deeb