**الترتيب sorting**

هي عملية ترتيب مجموعة من العناصر البيانية وفق قيمة حقل (او حقول) يسمى المفتاح (key)بصورة تصاعدية ascending)) أو بصورة تنازلية (descending)

**الغرض من الترتيب:**

تتعدد اغراض عملية الترتيب و أهمها :

* لزيادة كفاءة خوارزمية البحث عن عنصر
* لتبسيط معالجة الملفات
* لحل مشكلة تشابه القيود

**خطوات عملية الترتيب:-**

تتخلص خطواِت خوارزميات الترتيب بكافة أنواعها بالمراحل التالية:

* قراءة حقل المفتاح
* الاستدلال (استنتاج)موقع العنصر في الترتيب الجديد
* نقل العنصر البياني إلى موقعه الجديد

**أنوع خوارزميات الترتيب :-**

**1-الترتيب الداخليinternal sort :-**

وهو الترتيب الذي يحدث في الذاكرة الرئيسية للحاسوب main memory عندما يكون حجم البيانات مناسباً(ليس كبيراً) للخزن في الذاكرة ومن أهم أنواعه:

* الترتيب بالاختبار selection sort
* ترتيب الفقاعة sort(exchange)
* ترتيب الإضافة insertion sort
* الترتيب السريع quick sort
* ترتيب الأساس radix sort
* الترتيب الكومي heap sort
* ترتيب شيل shell sort

**2- الترتيب الخارجي external sort:**

وهو ترتيب البيانات المخزونة في أوساط الخزن الثانوية على شكل ملفات عندما يكون حجم البيانات كبير جداً بحيث يتعذر استيعابها كلها في الذاكرة في وقت واحد إثناء عملية الترتيب ومن أهم أنواعه:

* الترتيب بالدمج ذي المسارين two-way-merge sort
* الترتيب بالدمج متعدد المسارات k-way-merge sort
* الترتيب بالدمج المتوازن ذي المسارين balanced two-way-merge
* الترتيب بالدمج متعدد الأطوار poly phase two-way-merge

**العوامل الرئيسية المحددة لاختيار خوارزمية الترتيب:**

أن اختيار إي من خوارزميات الترتيب يجب أن يكون في ضوء عدد من العوامل من أهمها:

1. حجم البيانات المخزونة
2. نوع الخزن(الذاكرة الرئيسية , قرص , شريط)
3. درجة ترتيب البيانات(غير مرتبة , شبه مرتبة)

**ترتيب الفقاعة bubble sort:**

أن فكرة هذه الطريقة تتضمن اختيار اصغر القيم ووضعها في القائمة (أي أن القيمة الصغيرة تطفو على السطح)

وفيما يأتي خطوات الخوارزمية:

1. في المرحلة الأولى first pass :

العنصرين في الموقعين ((n ,(n-1) ونبادل موقعهما ليكون الأصغر قبل الأخر ،ونستمر إلى أعلى القائمة لحين الوصول إلى مقارنة العنصر في الموقع الثاني مع العنصر في الموقع الأول

1. في المرحلة الثانية(second pass) :

نقارن بنفس الطريقة السابقة ولكن من العنصر في الموقع (n) إلى العنصر في الموقع الثاني ( لان الموقع الأول اختير فيه العنصر الأقل قيمة في الخطوة السابقة - 1).

1. نكرر الخطوات أعلاه لـ(n-1) من المراحل .

**مثال**: لنقوم بترتيب القائمة تصاعديا 2,7,9,3,8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| القائمة الأصلية | المرحلة الأولى | المرحلة الثانية | المرحلة الثالثة | المرحلة الرابعة |
|  | 1 2 3 4 | 1 2 3 | 1 2 |  1 |
|  8 | 8 8 8 2 | 2 2 2 | 2 2  |  2 |
|  3 | 3 3 2 8 | 8 8 3 | 3 3  |  3 |
|  9 | 9 2 3 3 | 3 3 8  | 8 7 |  7 |
|  7 | 2 9 9 9 | 7 7 7 | 7 8  |  8 |
|  2 | 7 7 7 7 | 9 9 9 | 9 9  |  9 |

لاحظ أن العناصر في القائمة هو(n=5) وعدد المراحل (n-1=4)وعدد الخطوات في كل مرحلة يتناقص بمقدار واحد عن عدد خطوات المرحلة السابقة لها.

**ملاحظات:**

* معدل عدد المقارنات average no. of comparisons

(n2/2)حيث (n)يمثل عدد عناصر القائمة

* معدل عدد التبديلات average no. of exchangesهو (n2/4)
* هذه الطريقة جيدة عندما تكون العناصر شبه مرتبة وعددها ليس كبيرا ولا تحتاج إلى مساحة خزنيه كبيرة وبسيطة.
* أن وقت التنفيذ يبلغ (n2)O

**الترتيب بالاختيار selection sort :-**

وتتخلص خوارزمية هذا الترتيب بالخطوات التالية:

1-أيجاد اصغر عنصر في القائمة واستبداله من موقعه مع العنصر في الموقع الأول في القائمة

2-أيجاد اصغر عنصر في المتبقي من القائمة واستبداله من موقعه مع العنصر في الموقع الثاني من القائمة

3-نستمر في هذه العملية لحين الوصول إلى نهاية القائمة

**مثال:** رتب القائمة التالية تصاعديا

القائمة الأصلية 8 3 9 7 2 6 4

القائمة الأصلية 1 2 3 4 5 6

8 2 2 2 2 2 2

3 3 3 3 3 3 3

9 9 9 4 4 4 4

7 7 7 7 6 6 6

2 8 8 8 8 7 7

6 6 6 6 7 8 8

4 4 4 9 9 9 9

عدد عناصر القائمة n=7

عدد المراحل n-1=6 no. of passes

**ملاحظات:**

* معدل عدد المقارنات average no. of comparisons هوn\2\*(n-1)
* معدل عدد التبديلات average no. of exchanges هو ((n-1

**الترتيب بالإضافة:-**

تتخلص خطوات هذه الخوارزمية بما يأتي :

1- نبدأ بالعنصر الثاني ( I=2 ) في القائمة الأصلية ونقارنه مع العنصر الأول (I=1)

ونضعهم حسب الترتيب ( وليكن تصاعديا) في مقدمة القائمة .

2- نأخذ العنصر الثالث (I=3) في القائمة الاصلية ونقارنه مع مقدمة القائمة التي تحتوي العنصر الأول والثاني ونضعه في موقعه الصحيح معهم .

3- نأخذ العنصر الرابع (I=4) في القائمة الأصلية ونقارنه مع مقدمة القائمة التي تحتوي العناصر الثلاثة ونضعه في موقعه الصحيح بينهم .

4-نستمر في هذه العملية لغاية العنصر الاخير وسنحصل على القائمة مرتبة

مثال: نرتب عناصر القائمة التالية تصاعديا؟

القائمة الاصلية 1 2 3 4 5 6

8 ←3 3 3←2 2 2

3 8 8←7 3 3 3

9 9←9 8 7← 6← 4

7 7 7 9 8 7 6

2 2 2 2 9 8 7

6 6 6 6 6 9 8

4 4 4 4 4 4 9

عدد عناصر القائمةn=7

عدد المراحل n-1=6

اشارة السهم(←)توضح العنصر الذي أضيف

ملاحظات:

* معدل عدد المقارنات average of comparisonsهو(n/42)حيث(n) يمثل عدد عناصر القائمة
* معدل عدد التبديلات average no. of exchanges هو(n4/4)

**الترتيب السريع:-quick sort**

ان خوارزمية هذا الترتيب تعتمد فكرة التجزئة واللصق (divide & conquer)ففي ترتيب الفقاعة يتم مقارنة وتبديل العناصر المتجاورة لذلك اذا كان العنصر بعيد عن موقعه الصحيح ففي هذه الحالة سنحتاج إلى عدد كبير من المقارنات ان خوارزمية الترتيب السريع تعالج هذا الضعف وتسمح بأجراء المقارنات بين العناصر في المواقع المتباعدة وبأقل عدد من المقارنات إذ تعتمد فكرة التجزئة واللصق (divide & conquer)وتتلخص خطوات هذه الخوارزمية بالاتي:

1-اختيار احد عناصر القائمة في الوسط تقريبا وليكن ((xأي تقسيم القائمة إلى جزأين

2- يبدأ المسح من الاتجاهين أي نبحث في النصف الأول اليسار من القائمة عن العنصر الذي قيمته اكبر من(x)ونبحث في النصف الثاني (اليمين) من القائمة عن العنصر الذي قيمته اصغر من(x)نستبدل هذين العنصرين وذلك بجعل النصف الأول من القائمة يحتوي على عناصر اصغر من (x) ستبدل هذين العنصرين وذلك بجعل النصف الأول من القائمة يحتوي على عناصر اصغر من(x)والنصف الثاني يحتوي على عناصر اكبر من(x)

3- نأخذ النصف الأول من القائمة ونعالجه بنفس الأسلوب السابق (أي التجزئة واللصق) وهكذا مع النصف الثاني اي نستمر بالتجزئة واللصق تباعا لحين ترتيب جميع عناصر القائمة الكلية

مثال:-استخدم الخوارزمية للترتيب السريع (باعتماد العنصر الوسط محورا للترتيب)لترتيب مجموعة القيم التالية ترتيبا تصاعديا

(20,85,60,75,70,88,50,90,33,95)

الحل:-

أ/نفترض ان العناصر مخزونة المصفوفة (list)وبالصورة الاتية:-

F=1,l=10,x=lit[5]=70,i=1,j=10

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 33 | 90 | 50 | 88 | 70 | 75 | 60 | 85 | 20 |
| J |  |   |  |  |  |  |  |  | i |

ب/

I=2,j=9,list[2]<≠x,x<≠list[9],i=j

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 33 | 90 | 50 | 88 | 70 | 75 | 60 | 85 | 20 |

 J=9 Swap i=2

ج/

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 85 | 90 | 50 | 88 | 70 | 75 | 60 | 33 | 20 |

 ↑ ↑

 J=8 i=3

د/i=4,j=7,list[4]<≠x,x<≠list[7],i<=j

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 85 | 90 | 50 | 88 | 70 | 75 | 60 | 33 | 20 |

 ↑ swap ↑

 J=7 i=4

ه/

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 85 | 90 | 75 | 88 | 70 | 50 | 60 | 33 | 20 |

 ↑ ↑

 j i

و/

 i=5,j=5,list[5]<≠x,x<≠list[j]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 85 | 90 | 75 | 88 | 70 | 50 | 60 | 33 | 20 |

ij↑ ↑

عند هذه الخطوة جزئت القائمة إلى قسمين الأيسر يحتوي على جميع الأعداد التي قيمتها اقل من(70) والقسم الأيمن يحتوي على جميع الأعداد التي قيمتها اكبر من (70) أما الخطوة التالية فهي تنفذ الخوارزمية بصورة متكررة على جزء بنفس الطريقة أي استدعاء (1,4)quick sort 1 فيما يتعلق بالجزاء الذي عناصره في المواقع من(6)إلى (10)وهكذا تستمر عملية تكرار التجزئة والترتيب

طريقة أخرى:-

في الخطوات السابقة نلاحظ اختيار العنصر الواقع وسط القائمة ليكون محور (مركز)المقارنة ليكون(pivot)ونقارن معه العناصر الأخرى وهذه طريقة أخرى تتضمن اختيار العنصر في الموقع الأول لهذا الغرض وبموجب خطوات الخوارزمية الاتية :-

1-اختيار العنصر في الموقع الأول ليكون محور (pivot)التجزئة

2-نقل هذا العنصر وأخلاء موقعه

3-نبدأ مسح العناصر من الجهة الأخرى أي اليمين ونقارن كل عنصر مع العنصر المحور(pivot value)

4-عند أيجاد عنصر اصغر من العنصر المحور ينقل ذلك العنصر إلى الموقع الذي كان فيه العنصر المحور ويبقى موقعه خاليا

5-نمسح العناصر من جهة اليسار باتجاه اليمين ونقارن هذه العناصر مع العنصر المحور فإذا وجدنا عنصرا اكبر منه ننقله إلى الموقع الخالي ويترك موقعه خاليا

6-ننتقل إلى جهة اليسار لنمسح العناصر باتجاه اليمين لحين الوصول إلى عنصر اكبر من العنصر المحور وننقله إلى الجهة الأخرى وبنفس الأسلوب ننتقل إلى الجهة اليمنى

7- بعد توزيع العناصر التي اكبر من العنصر المحور في اليمين والعناصر التي اصغر منه في اليسار نعيد العنصر المحور إلى الموقع الخالي ليصبح هو الفاصل بينهما

8- نكرر الخطوات السابقة على عناصر القائمة عدا العنصر الأول ثم نكرر مرة أخرى على القائمة عداد العنصرين الأولين وهكذا لحين انتهاء عملية الترتيب

مثال:- استخدام الخوارزمية quickort2 للترتيب السريع (باعتماد العنصر الأول محورا للترتيب ) لترتيب مجموعة القيم الاتية ترتيبا تصاعديا (50,85,10,75,38,90,30,70,40,95)

الحل:-

ٲ/نفترض ان العناصر مخزونة في المصفوفة list وبالصورة الاتية:-

F=1,l=10,i=1,j=10,x=list[1]=50

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 40 | 70 | 30 | 90 | 38 | 75 | 10 | 85 | 50 |

↑J ↑i

ب/

 i=1,j=9list[9]>≠x

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 40 | 70 | 30 | 90 | 38 | 75 | 10 | 85 | 50 |

 ↑ i=1

 j ← move it←←

ج/

i=1,j=9,list[i]=list[j]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 40 | 70 | 30 | 90 | 38 | 75 | 10 | 85 | 40 |

 ↑ ↑

 j i

د/

I=2,j=9,list[i]<≠=x

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 40 | 70 | 30 | 90 | 38 | 75 | 10 | 85 | 40 |

 ↑ ↑

 J move it →→

ه/

i=2,j\=9,list[j]=list[i]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 85 | 70 | 30 | 90 | 38 | 75 | 10 | 85 | 40 |

 ↑ ↑

 j i

و/

I=2,j=7,list[j]>≠x

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 85 | 70 | 30 | 90 | 38 | 75 | 10 | 85 | 40 |

 ↑ ↑

 j←←move it← i

ز/

I=2,j=7,list[i]=list[j]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 85 | 70 | 30 | 90 | 38 | 75 | 10 | 30 | 40 |

 ↑ ↑

 j i

ح/

I=4,j=7,list[i]<≠x

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 85 | 70 | 30 | 90 | 38 | 75 | 10 | 30 | 40 |

 j ↑ ↑ i

 move it →→

ط/

I=4,j=7,list[j]=list[i]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 85 | 70 | 75 | 90 | 38 | 75 | 10 | 30 | 40 |

 ↑ ↑

 j i

ك/

I=4,j=5,list[j]>≠x

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 85 | 70 | 75 | 90 | 38 | 75 | 10 | 30 | 40 |

 j ↑ ↑i

 ←←move it

ل/

I=4,j=5,list[i]=list[j]

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 85 | 70 | 75 | 90 | 38 | 75 | 10 | 30 | 40 |

 j↑ ↑ i

 م/

I=5,j=5

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 85 | 70 | 75 | 90 | 38 | 38 | 10 | 30 | 40 |

J ↑↑i

ن/

I=5,j=5,i<≠j,list[i]=x

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
| 95 | 85 | 70 | 75 | 90 | 50 | 38 | 10 | 30 | 40 |

j ↑↑i

عند هذه الخطوة جزئت إلى قسمين القسم الأيسر يحتوي على جميع الأعداد التي قيمتها اقل من(50) والقسم الأيمن يحتوي على جميع الأعداد التي قيمتها اكبر من(50) في الخطوة التالية تنفذ نفس الخوارزمية على الجزاء الأيسر أي باستدعاء (f,i-1) quicksort2 ثم على الجزاء الأيمن لحين ترتيب جميع عناصر القائمة ان تحليل خوارزمية الترتيب السريع يشير إلى ان معدل عدد المقارنات (average no. Of comparisons)هو (n log 2 n)ومعدل عدد التبديلات (average no. of exchanges)هو(n log2 n2/1) وان استخدام هذه الخوارزمية لترتيب عناصر قائمة مرتبة سيتطلب وقت تنفيذ طويل يتناسب مع(2n)بدلا من المعدل(n log2 n)

**ترتيب الدمج:- balanced two-way merge**

هذه الطريقة هي من أنواع الترتيب الخارجي وتتلخص الخوارزمية بالخطوات الاتية :

1. تقسيم القائمة (البيانات)الاصلية إلى قائمتين متساويتين تقريبا ولتكن a,b
2. نقارن العنصر الأول من القائمةa مع نظيره العنصر الأول من القائمة bونضعهما بالترتيب في القائمةd
3. نقارن العنصر الثاني من القائمةa مع نظيره العنصر الثاني من القائمة bونضعهما بالترتيب في القائمة d
4. نكرر الخطوتين 2 و3 وسنحصل على عناصر مزدوجة (string of length 2)في كل من القائمتينd,c
5. بنفس الطريقة نقوم بدمج عناصر القائمتينd,c ونضعهما في القائمتين b, a وستكون عناصرهما بطول (string of 4)
6. نعيد الطريقة بدمج عناصر القائمتين b,aونضعهما في القائمتينd,c وستكون عناصرهما بطول (string of 8)
7. نستمر بهذا الأسلوب لحين الحصول على القائمة النهائية المرتبة

مثال :-

نرتب تصاعديا القائمة أدناه باستخدام خوارزمية الدمج(merge sort)

3و47و33و28و20و63و42و50و02و23و18

الحل:-

ان عدد عناصر القائمة (n=11)وتجزئ إلى قائمتين متساويتين بالعدد تقريبا

 A:18,23,2,50,42

B:63,20,28,33,47,3

C:18,63,2,28,42,47

D:20,23,33,50,3

A:18,20,23,63,3,42,47

B:2,28,33,50

C:2,18,20,23,28,50,63

D:3,42,47

A:2,3,18,20,23,28,42,47,50,63

بعد سلسلة الخطوات أعلاه حصلنا على عناصر القائمة مرتبة تصاعديا كما في (a)