خوارزميات البحث

المحاضرة الرابعة
م. لمى السبع

خوارزمية البحث الخطي

- خوارزمية البحث الخطي هي إحدى خوارزميات البحث التقليدية و الأساسية، إذ تُعتبر طريقة للبحث عن **موقع** قيمة معينة داخل مصفوفة، و أساسيتها تنبع من إتباعها منهجية بسيطة جداً، فهي تُنجز عملية البحث بالتحقق من كل عنصر بصورة متسلسلة أو كما يروق لمتخصصي الخوارزميات (خطية). تبدأ من بداية المصفوفة و حتى نهايتها.

طريقة عمل خوارزمية البحث الخطي

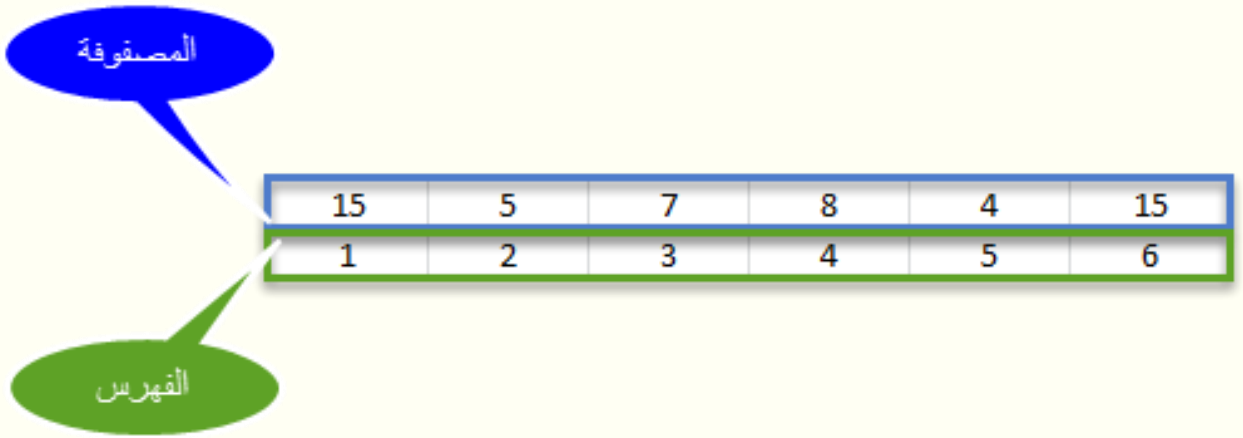
- لإجراء البحث الخطي لا بد من توفر عناصر أساسية و هي مصفوفة البحث و العنصر المُراد البحث عنه.
- بعكس كثير من خوارزميات البحث، لا يتطلب أن تكون مصفوفة البحث مرتبة ترتيباً تصاعدياً أو تنازلياً، هذا لا يعني أن المصفوفة إذا كانت مرتبة فإنها تُخل بأساسيات البحث الخطي، على العكس تماماً، فكل أنواع المصفوفات مرتبة أو غير مرتبة مُرحبٌ بها في خوارزمية البحث الخطي.
- مصفوفة البحث لها فهرس (Index)، يبدأ من الواحد أو الصفر، لا تختلف كثيراً فالنتيجة واحدة.
- يبدأ البحث بالتحقق من العنصر الأول هل هو مساوٍ للعُنصر المراد البحث عنه، إذا وُجد العنصر المراد البحث عنه يُحتفظ برقم الفهرس الخاص بالعنصر و تُنهى عملية البحث.
- إذا لم يُوجد العنصر المراد البحث عنه تنتقل عملية البحث من العنصر الأول في المصفوفة إلى العنصر الثاني ثم الثالث و هكذا إلى أن يكتمل البحث عن العُنصر المراد البحث عنه في كامل المصفوفة.
- يُرمز لحالة عدم إيجاد العُنصر المراد البحث عنه بقيمة مختلفة عن قيم فهرس المصفوفة، مثلاً إذا كانت المصفوفة بطول ١٠ عناصر، و تبدأ من ١ إلى ١٠، فيُرمز إلى حالة عدم إيجاد العنصر المراد البحث عنه بالرقم (٠)، لاحظ أنه كما ذُكر سابقاً في النقطة رقم ٣، في بعض المصفوفات قد يبدأ الفهرس من الرقم **صفر**، في هذه الحالة قد يُلجأ إلى الإشارة بالقيمة NULL في حالة عدم إيجاد العنصر المراد البحث عنه.

مثال لطريقة البحث في خوارزمية البحث الخطي

- نفترض أن لدينا المصفوفة أدناه حيث :

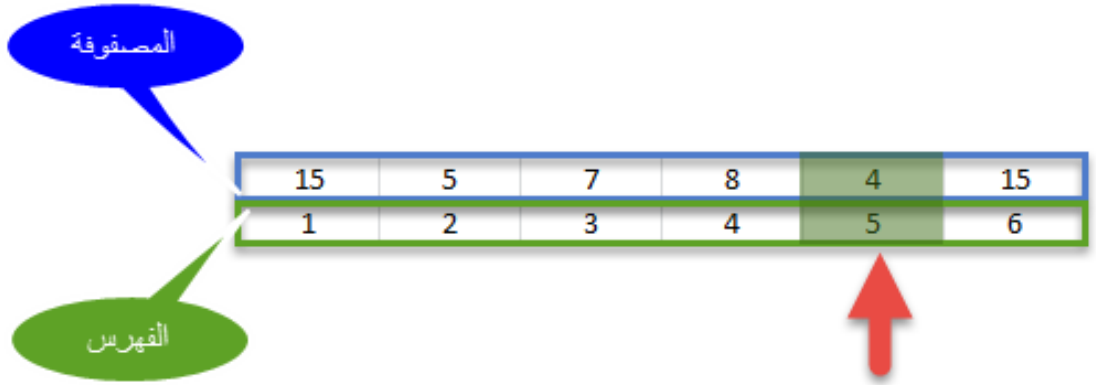
❖عدد عناصر المصفوفة n=6

❖القيمة المراد البحث عنها x=4



آلية الحل:

- تبدأ عملية البحث بمقارنة العنصر بالخانة رقم 1 حيث j=1، هل A[j]=x؟ لا. إذن ننتقل إلى الخطوة الثانية.
- مقارنة العنصر بالخانة رقم 2 حيث j=2، هل A[j]=x؟ لا. إذن ننتقل إلى الخطوة الثالثة.
- مقارنة العنصر بالخانة رقم 3 حيث j=3، هل A[j]=x؟ لا. إذن ننتقل إلى الخطوة الرابعة.
- مقارنة العنصر بالخانة رقم 4 حيث j=4، هل A[j]=x؟ لا. إذن ننتقل إلى الخطوة الخامسة.
- مقارنة العنصر بالخانة رقم 5 حيث j=5، هل A[j]=x؟ نعم. إذن هنا نحتفظ بالرقم 5 و تُعتبر هذه نتيجة البحث.



الكود البرمجي

خوارزمية البحث الثنائي Binary Search

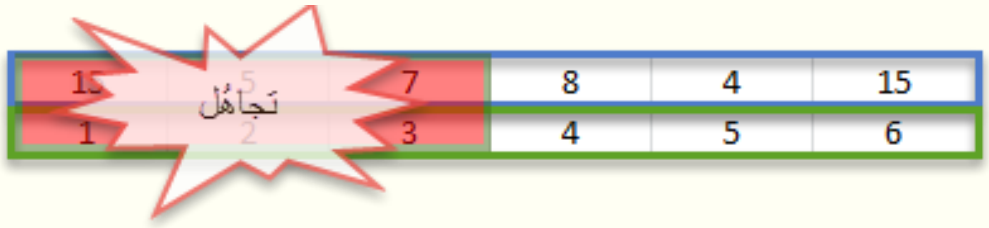
- كغيرها من الخوارزميات لا بد من توافر المدخلات المطلوبة و التي تتمثل في العُنصر المراد البحث عنه و المصفوفة التي سيتم البحث فيها
- يجب أن تكون المصفوفة مرتباً تصاعدياً (من الأصغر إلى الأكبر)، في حال لم تكن المصفوفة مرتبة تصاعدياً يجب عليك أن تُرتبها تصاعدياً حتى تستطيع تطبيق خوارزمية البحث الثنائي
- مصفوفة البحث لها فهرس ((Indexيبدأ من الصفر أو الواحد، لكليهما نفس النتيجة

آلية عمل الخوارزمية

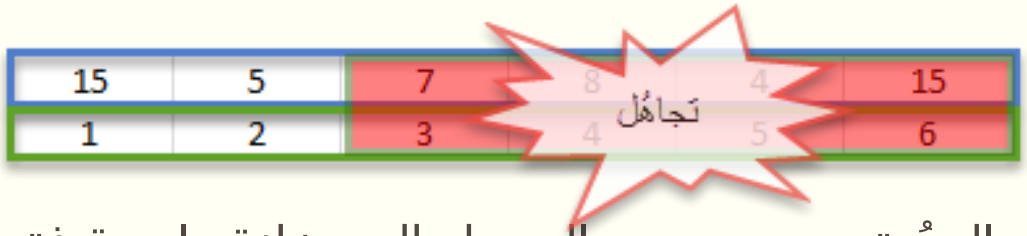
- بدأ الخوارزمية عملها بوضع مؤشر الفهرس في وسط المصفوفة حتى تقسم المصفوفة إلى قسمين، و هنا بداية الاستفادة الفعلية من مبدأ (فرّق تسُد)
- تتم مقارنة العنصر بالوسط (حيثُ المؤشر) بالعُنصر المُراد البحث عنه و النتيجة إحدى ثلاث حالات:

١. أن تساوي قيمة العُنصر بالمصفوفة قيمة العُنصر المُراد البحث عنه، و بالتالي يتم الإحتفاظ بقيمة مؤشر الفهرس و يُوقف البحث.

٢. ن تكون قيمة العُنصر المراد البحث عنه أكبر من قيمة العنصر حيث المؤشر، في هذه الحالة يتم تجاهل جميع عناصر النصف الأيسر من المصفوفة و يُصبح الجزء الأيمن من المصفوفة مُدخلاً جديداً لعملية بحث عن طريق خوارزمية البحث الثنائي.

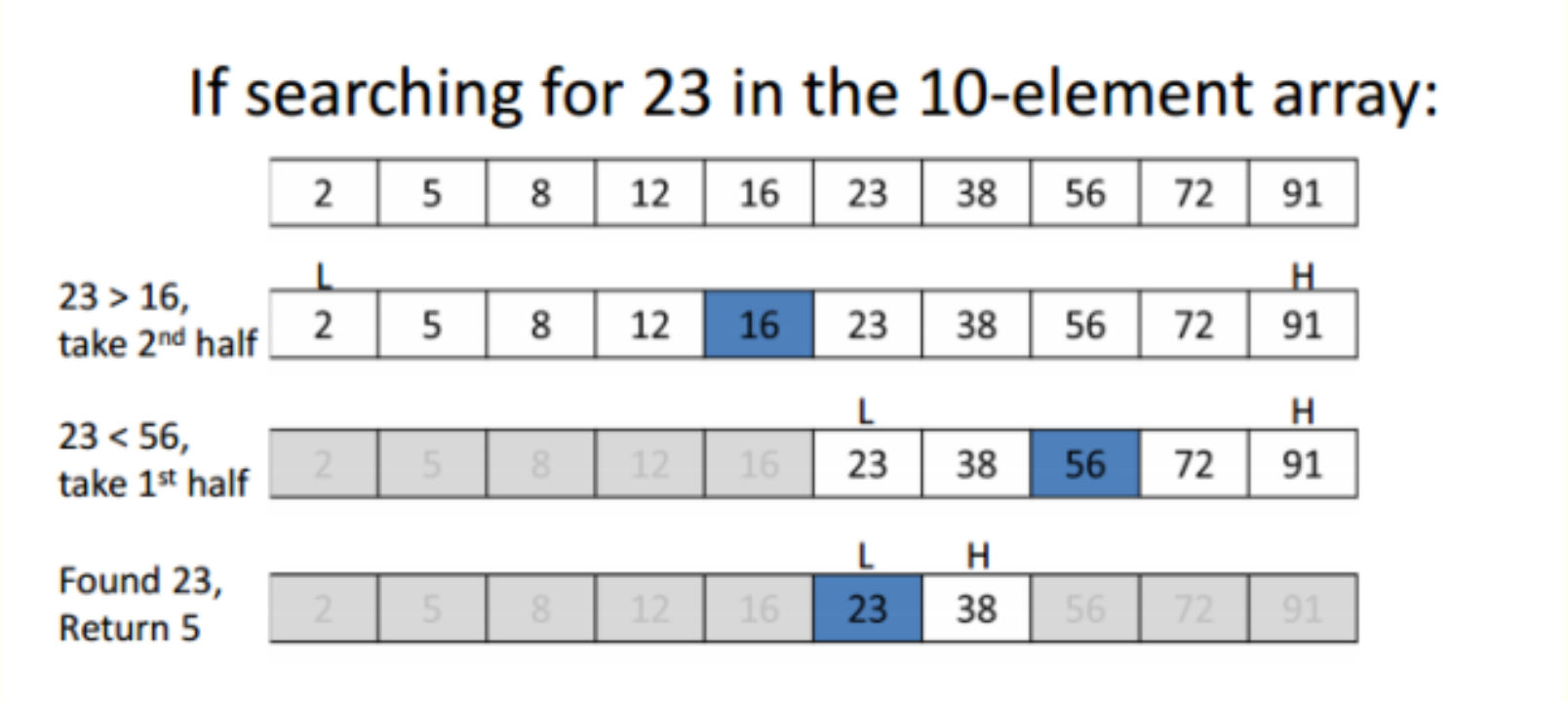


٣. أن تكون قيمة العُنصر المراد البحث عنه أقل من قيمة العنصر حيثُ المؤشر، في هذه الحالة يتم تجاهل جميع عناصر النصف الأيمن من المصفوفة و يُصبح الجزء الأيسر من المصفوفة مُدخلاً جديداً لعملية بحث عن طريق خوارزمية البحث الثنائي



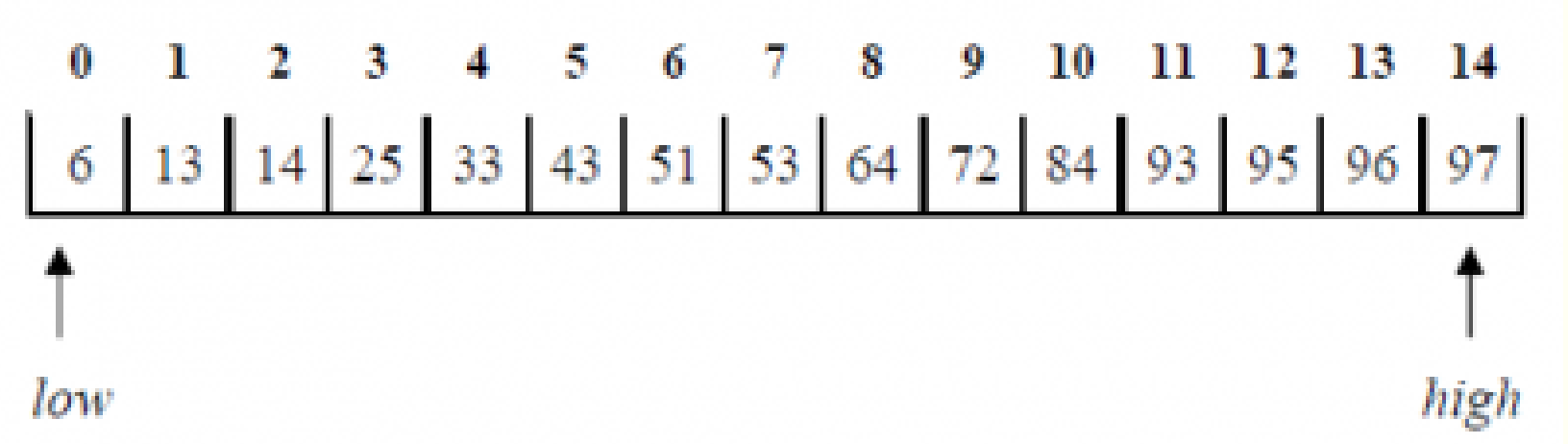
- تُكرر عملية البحث في النصف المتُبقي حتى يتم الوصول إلى خانة واحدة فقط تُمثل موقع العنصر المُراد البحث عنه

مثال


If searching for 23 in the 10-element array:
2 5 8 12 16 23 38 56 72 91
23 > 16, take 2nd half
L H
2 5 8 12 16 23 38 56 72 91
23 < 56, take 1st half
L H
2 5 8 12 16 23 38 56 72 91
Found 23, Return 5
L H
2 5 8 12 16 23 38 56 72 91

مثال آخر

المطلوب : البحث عن القيمة ٣٣

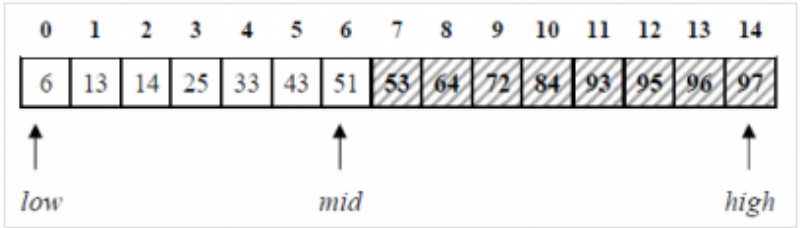


1. نحدد منتصف المصفوفة:

$$Mid = \frac{low + high}{2} = \frac{(0 + 14)}{2} = 7$$

2. تقسيم المصفوفة إلى قسمين وفقا لموقع mid:

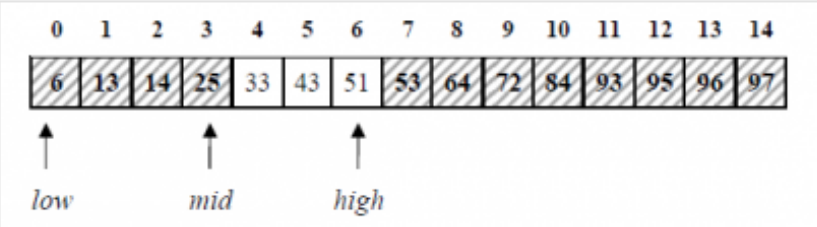
3. بما أن 33 أصغر من 53, فعملية البحث ستنفذ على النصف الأول:



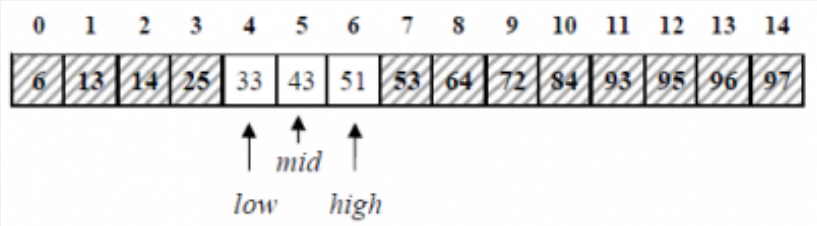
4. نعيد عملة التقسيم على النصف الأول و تصبح قيمة mid تساوي:

$$Mid = \frac{(0 + 7)}{2} = 3$$

5. بما أن 33 أكبر من 25, سيتم استبعاد النصف الأول و تستمر عملية البحث في النصف الثاني:



6. من جديد, نعيد عملية التقسيم حيث تصبح قيمة mid تساوي 5:



7. و أخيرا نجد أن القيمة 33 موجوده في الموقع low و الذي يساوي 4.

الكود البرمجي

```cpp
#include <iostream>
using namespace std;

int binarySearch(const int array[], int key, int low, int high) {
    int mid;
    if (high < low) return -1;
    else {
        mid = (low + high) / 2;
        if (array[mid] > key)
            return binarySearch(array, key, low, mid - 1);
        else if (array[mid] < key)
            return binarySearch(array, key, mid + 1, high);
        else
            return mid;
    }
}

int main() {
    const int LENGTH = 1000, centre = LENGTH / 2;
    int result, X[LENGTH];
    for (int i = 0, j = -centre; j <= centre; i++, j++)
        X[i] = j;
    result = binarySearch(X, -500, 0, LENGTH - 1);
    if (result != -1)
        cout << "la clef se trouve dans la case numéro " << result << endl;
    else
        cout << "Valeur introuvable" << endl;
    return 0;
}
```