

Search Algorithm

خوارزميات البحث

إذا كان قائمة من البيانات مخزونة في مصفوفة أو في ملف فالبحث هو طريقة الوصول إلى تلك البيانات المخزونة في المصفوفة أو الملف واسترجاعها عند الحاجة إليها، وتختلف طريقة البحث باختلاف طريقة خزن البيانات ومن طرق البحث:

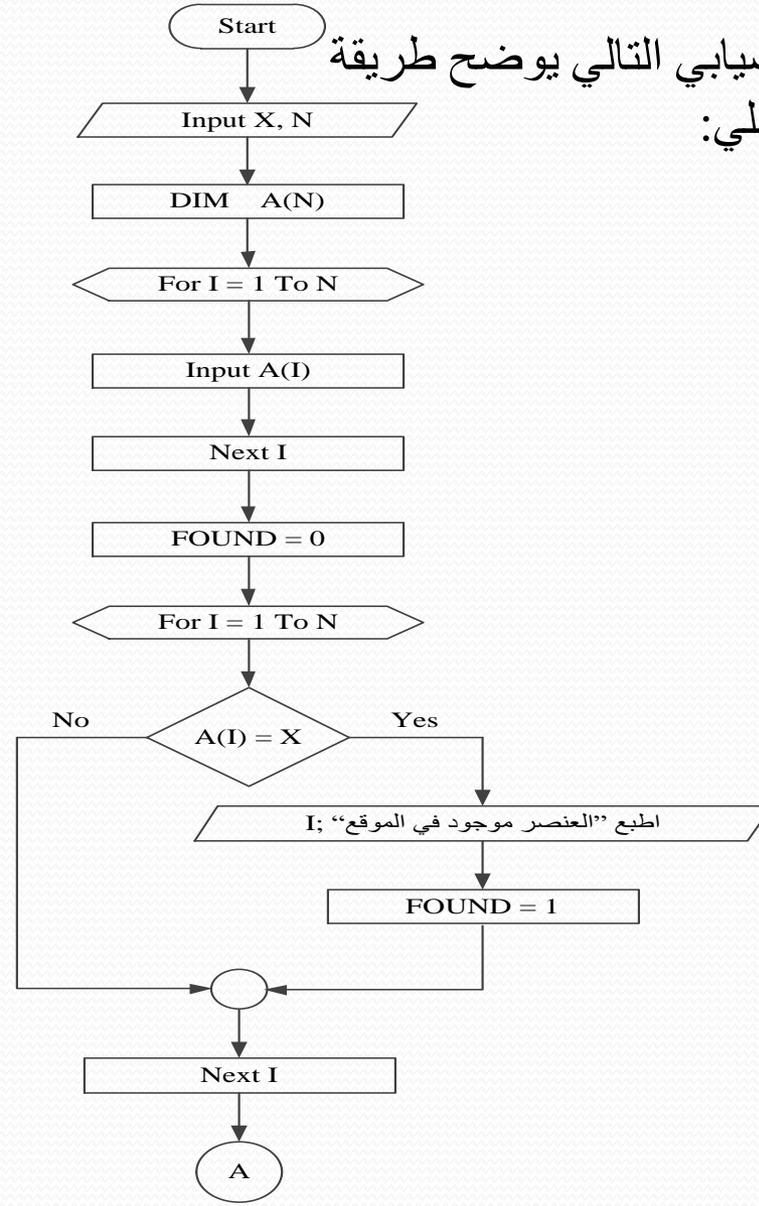
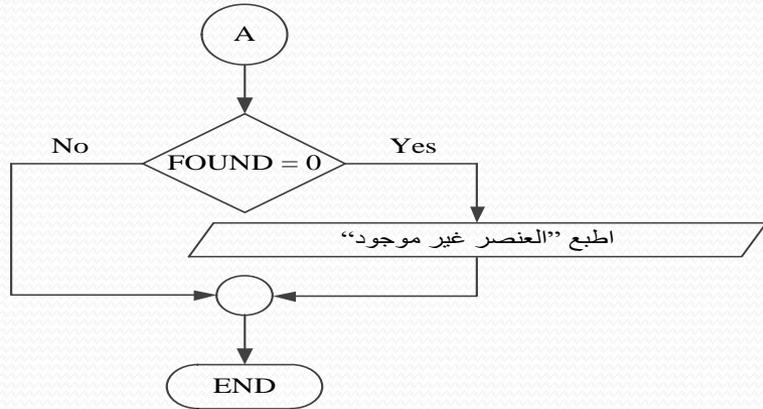
أولاً: البحث المتسلسل (البحث الخطي) Sequential Search

يعتبر هذا النوع من أبسط طرق البحث وتستخدم هذه الطريقة للبحث في مجموعة البيانات غير المرتبة ويتم البحث التسلسلي بمقارنة العنصر المراد البحث عنه ابتداءً بالعنصر الأول ثم الثاني ثم الثالث وهكذا إلى العنصر الأخير لمجموعة البيانات (المصفوفة) ويعتبر هذا النوع من البحث مناسب لأسلوب الخزن التسلسلي كالأشرطة المغناطيسية.

ملاحظة:

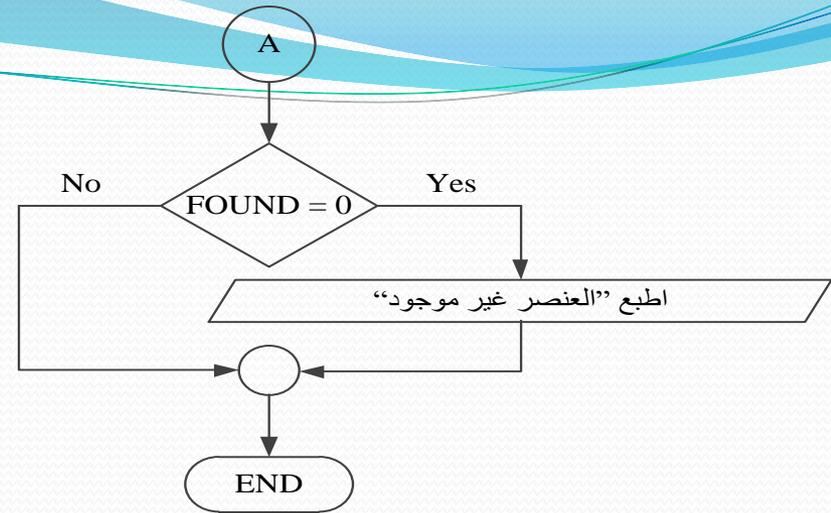
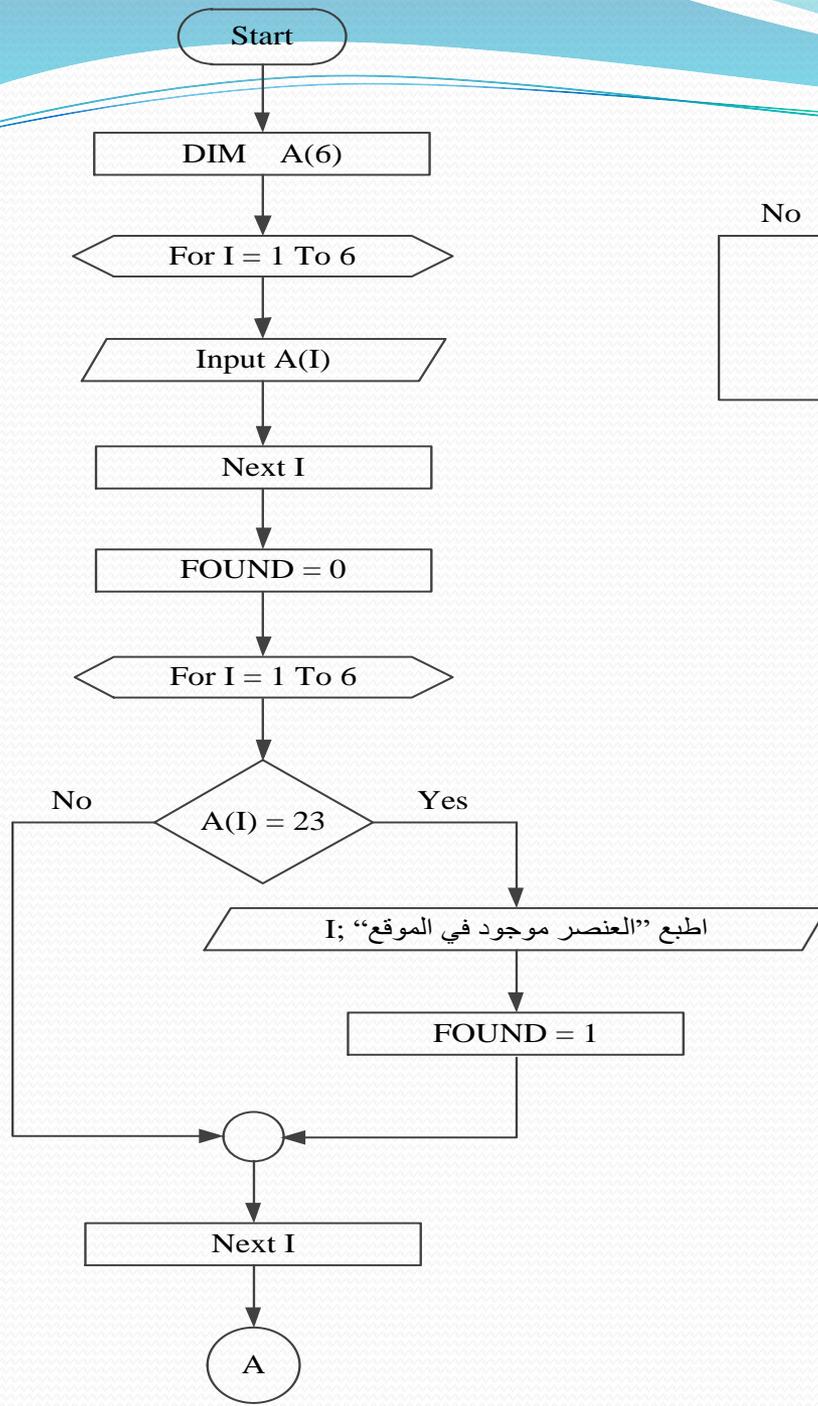
يجب أن يكون العنصر المطلوب البحث عنه من النوع نفسه للمصفوفة.

المخطط الانسيابي التالي يوضح طريقة
البحث التسلسلي:



ملاحظة:

صيغة أخرى للسؤال:
مصفوفة مكونة من N من القيم العددية ارسم مخطط انسيابي
للبحث عن قيمة معينة ضمن هذه المصفوفة، مستخدماً البحث
التسلسلي في الحل.



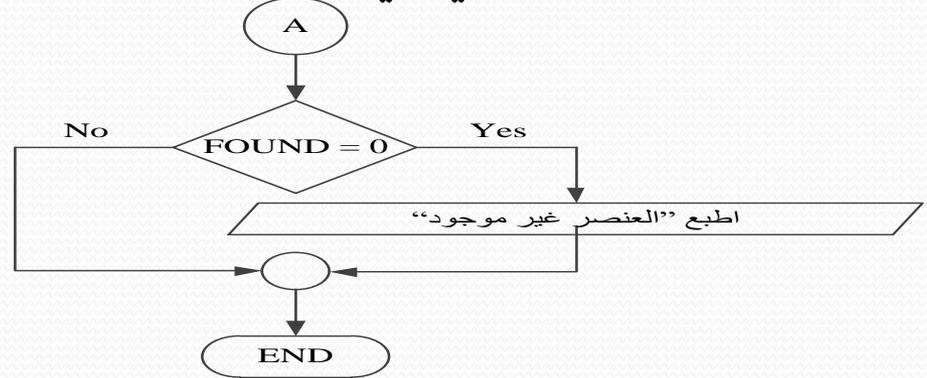
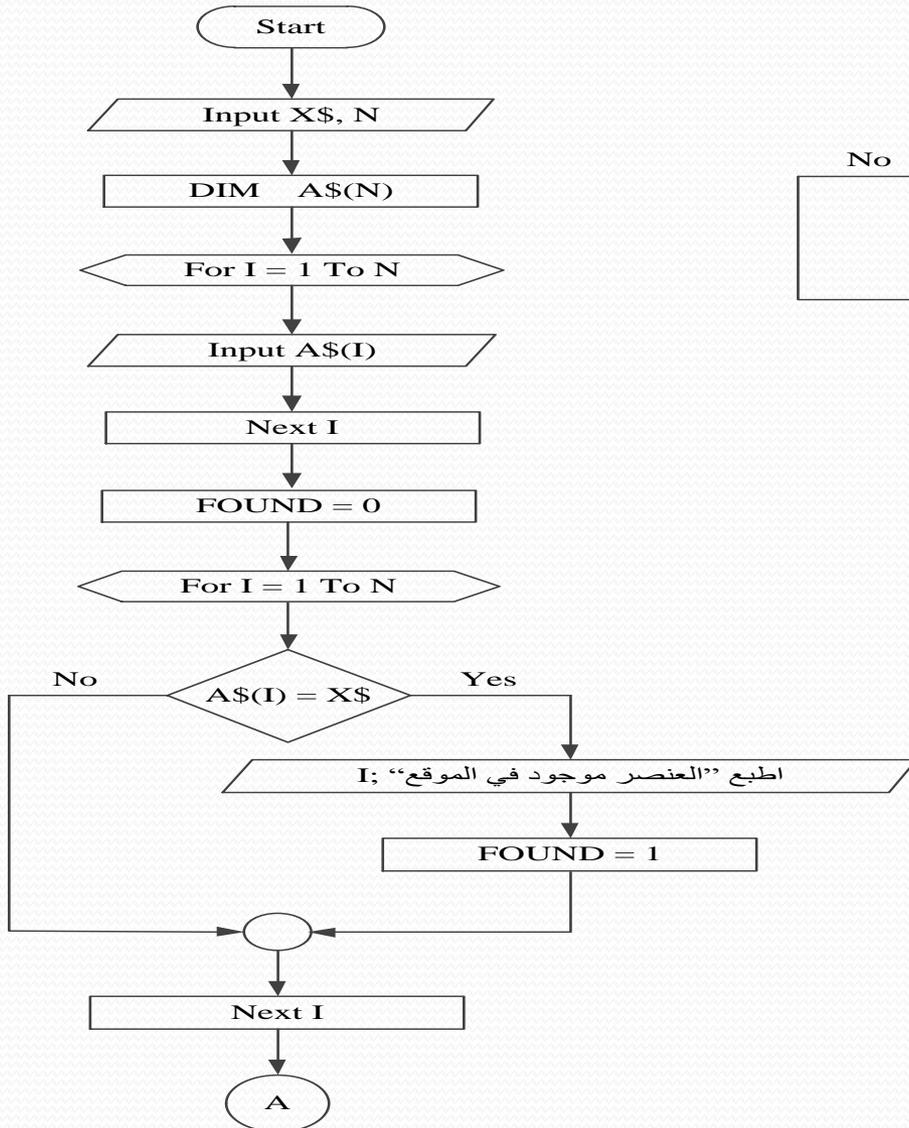
مثال:

مصفوفة مكونة من 6 عناصر، ارسم مخطط انسيابي للبحث عن القيمة (23) ضمن عناصر المصفوفة مستخدماً البحث التسلسلي في الحل علماً أن عناصر المصفوفة هي كالآتي:

[17, 23, 26, 43, 23, 75]

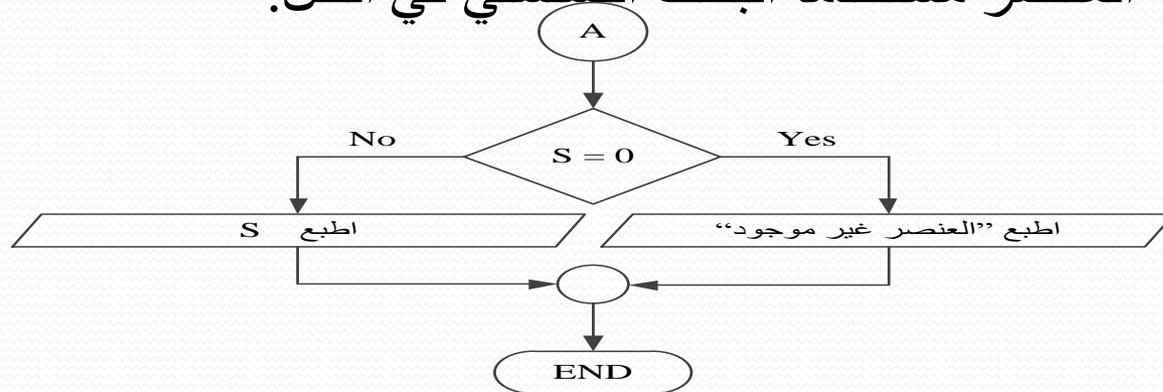
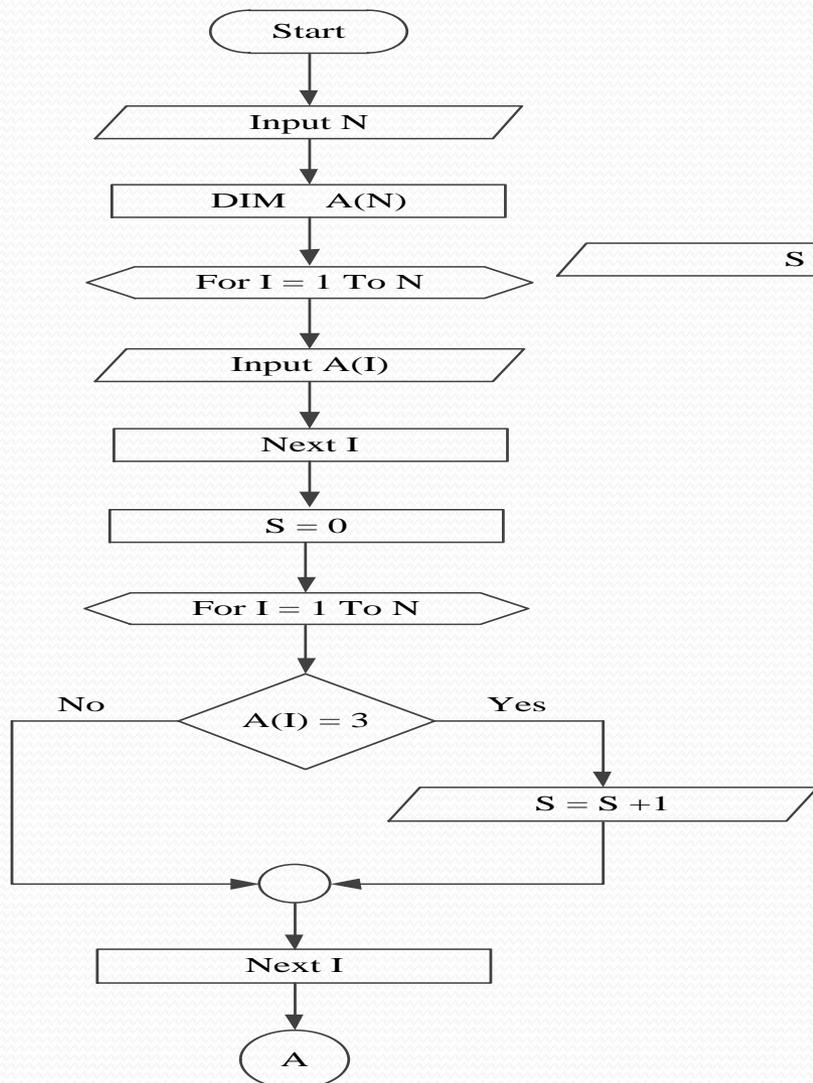
مثال:

ارسم مخطط انسيابي للبحث عن موقع اسمك ضمن قائمة اسماء لطلبة شعبتك مستخدماً البحث التسلسلي في الحل.



مثال:

ارسم مخطط انسيابي لحساب عدد مرات ظهور الرقم (3) ضمن عناصر المصفوفة $A(N)$ وطباعة عبارة (العنصر غير موجود) في حالة عدم وجود العنصر مستخدماً البحث التسلسلي في الحل.



ثانياً: البحث الثنائي Binary Search

- هي طريقة للوصول إلى بيانات معينة في جدول البيانات المرتبة (حسب التسلسل العددي تصاعدياً أو تنازلياً إذا كانت البيانات عددية أو حسب التسلسل الابددي إذا كانت البيانات حرفية) وتعتبر طريقة البحث الثنائي أفضل طرق البحث حيث أن عملية الوصول إلى القيمة المطلوبة تكون أسرع.
- وتستخدم هذه الطريقة من البحث (البحث الثنائي) عندما يصادف المبرمج التعامل مع كمية كبيرة من البيانات.

ولكي نطبق خوارزمية البحث الثنائي (Binary Search) على أي مصفوفة نتبع الخطوات التالية:

1. الخطوة الأولى والتي يمكن اعتبارها أهم خطوة والتي لا يمكن تطبيق البحث الثنائي (Binary Search) لولاها هي: ترتيب عناصر المصفوفة تصاعدياً أو تنازلياً أو ابددياً حسب نوع البيانات المخزنة فيها.
2. تحديد موقع أول عنصر في المصفوفة وليكن I، وموقع آخر عنصر في المصفوفة وليكن N.
3. تحديد موقع العنصر الذي يقع في منتصف المصفوفة وليكن L.

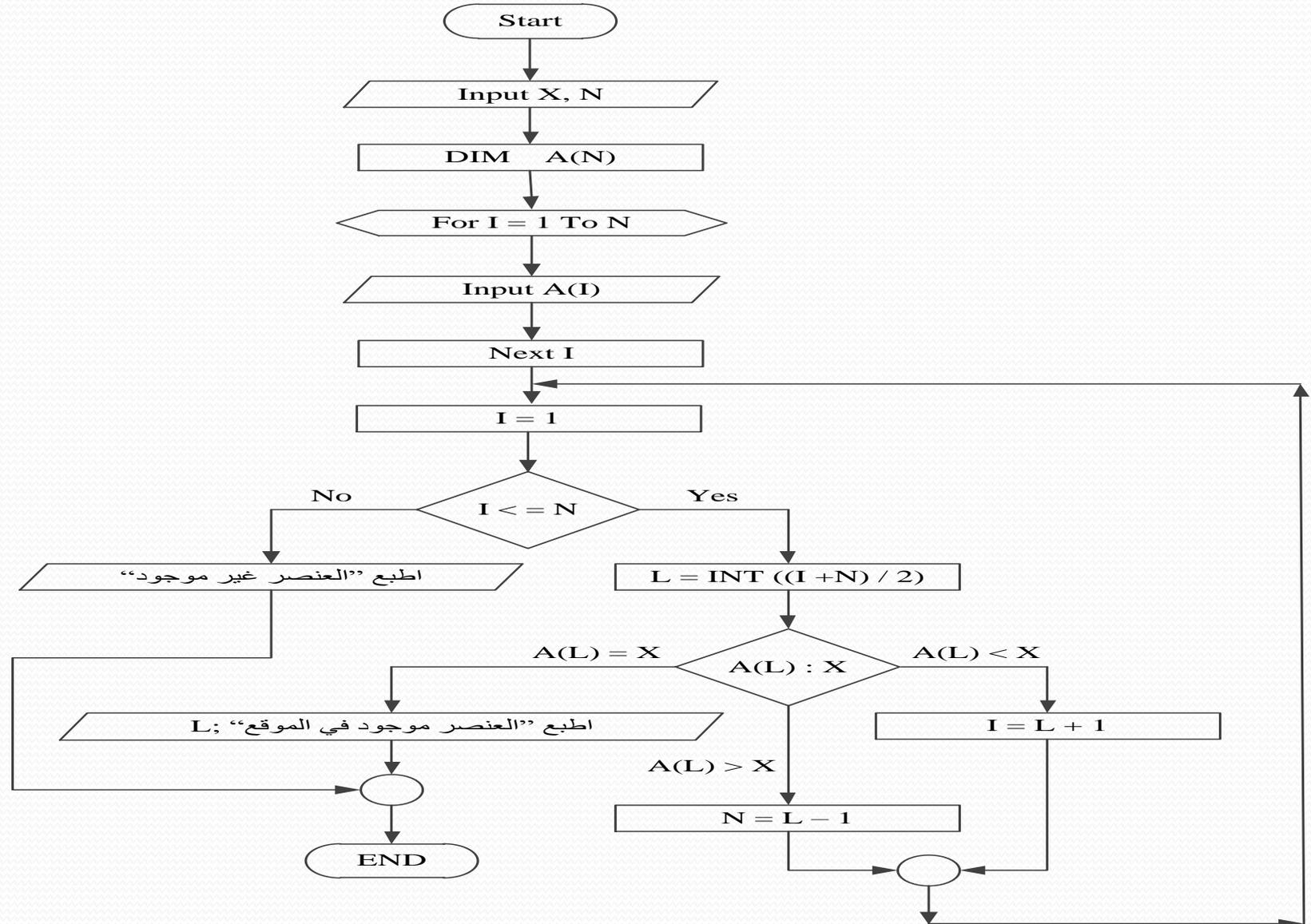
تقوم فكرة البحث الثنائي على أساس تقسيم المصفوفة إلى نصفين واستبعاد النصف الذي لا ينتمي إليه العنصر المراد البحث عنه، أي عن طريق تحديد العنصر الذي يقع في منتصف المصفوفة، ثم نقارن العنصر المراد البحث عنه كالاتي (لنفرض أن المصفوفة مرتبة تصاعدياً):

1. إذا كان العنصر المراد البحث عنه يساوي العنصر الموجود في منتصف المصفوفة (العنصر الوسطي) نكون قد وجدنا العنصر الذي نبحث عنه.
2. إذا كانت قيمة العنصر المراد البحث عنه أقل من قيمة العنصر الوسطي في المصفوفة، نبحث فقط في نصف المصفوفة الأول (العلوي) ونستبعد البحث في نصفها الثاني (السفلي).
3. إذا كانت قيمة العنصر المراد البحث عنه أكبر من قيمة العنصر الوسطي في المصفوفة، إذن نحتاج أن نبحث فقط في نصف المصفوفة الثاني (السفلي) ونستبعد البحث في نصفها الأول (العلوي).
4. نعتبر النصف الذي حددنا البحث فيه مصفوفة قائمة بحد ذاتها، نحدد فيها الـ I ، N ، أي نقوم بتقسيمها إلى قسمين) ونطبق نفس الخطوات من 1 إلى 3 فيها، ثم نقارن العنصر المراد البحث عنه مع العنصر الوسطي الجديد، بنفس الترتيب الذي ذكر في الخطوات من 1 إلى 3 السابقة.

وللحصول على موقع الوسط لجدول البيانات (المصفوفة) نستخدم المعادلة التالية:

$$L = \text{INT} ((I + N) / 2)$$

المخطط الانسيابي التالي يوضح طريقة البحث الثنائي Binary Search



ملاحظة:

صيغة أخرى للسؤال:

ارسم مخطط انسيابي للبحث عن قيمة عددية معينة ضمن مصفوفة تتكون من N من القيم العددية المرتبة تصاعدياً مستخدماً البحث الثنائي.

صيغة أخرى للسؤال:

ارسم مخطط انسيابي للبحث عن العدد (7) ضمن عناصر المصفوفة A حيث أن:

$$A = [1, 2, 4, 6, 7, 10]$$

مستخدماً أفضل طريقة في البحث.

مثال:

ارسم مخطط انسيابي للبحث عن العدد (13) ضمن عناصر المصفوفة A حيث أن:

$$A = [13, 37, 50, 10, 10, 70, 8, 43]$$

مستخدماً طريقة البحث الثنائي (أو مستخدماً أفضل طريقة في البحث) في الحل.

ملاحظة مهمة:

نلاحظ البيانات (أو القيم العددية) غير مرتبة فيجب أن ترتب أولاً ترتيباً تصاعدياً بأحد طرق الترتيب ثم بعد الترتيب نبدأ البحث الثنائي.

