

Binary search Algorithm

Binary search is a search algorithm that is used to find elements in a sorted list.

The algorithm starts by examining the element in the middle of the list. If the value of the element is equal to the value that is being searched for, the algorithm stops. If the value of the element is less than the value that is being searched for, the algorithm searches the right half of the list. And if the value of the element is greater than the value that is being searched for, the algorithm searches the left half of the list. The algorithm continues to divide the list in half until it finds the element or until the list is empty.

Binary search is a very efficient search algorithm, as it can find the element in a list of size n in time $O(\log n)$.

Here is an example of binary search on the array [1, 3, 5, 7, 9] to find the target value 5:

Start with the leftmost element of the array (index 0).

The middle element of the array is 5 (index $(0 + \text{len}(\text{arr})) / 2$).

Since x is equal to the middle element, return the index of the middle element, which is 2.

Therefore, the binary search algorithm will return the value 2.

Binary search is a very efficient algorithm for searching sorted data. The worst-case time complexity of binary search is $O(\log n)$, where n is the number of elements in the array. This means that the time it takes to search the array grows logarithmically with the number of elements, which is much better than the linear time complexity of a simple search algorithm.

Binary search is a fundamental algorithm in computer science and is used in a variety of applications, such as searching for files in a filesystem, finding records in a database, and sorting data.

خوارزمية البحث الثنائي

البحث الثنائي هو خوارزمية بحث تستخدم للعثور على عناصر في قائمة مرتبة.

تبدأ الخوارزمية بفحص العنصر في منتصف القائمة. إذا كانت قيمة العنصر تساوي القيمة التي يتم البحث عنها ، فإن الخوارزمية تتوقف. إذا كانت قيمة العنصر أقل من القيمة التي يتم البحث عنها ، فإن الخوارزمية تبحث في النصف الأيمن من القائمة. وإذا كانت قيمة العنصر أكبر من القيمة التي يتم البحث عنها ، فإن الخوارزمية تبحث في النصف الأيسر من القائمة. تستمر الخوارزمية في تقسيم القائمة إلى نصفين حتى تجد العنصر أو حتى تكون القائمة فارغة.

البحث الثنائي هو خوارزمية بحث فعالة للغاية ، حيث يمكنه العثور على العنصر في قائمة من حجم n في وقت $O(\log n)$.

فيما يلي مثال على البحث الثنائي في المصفوفة [1، 3، 5، 7، 9] للعثور على القيمة المستهدفة 5:

- ابدأ بالعنصر الأيسر في المصفوفة (المؤشر 0).
 - العنصر الأوسط من المصفوفة هو 5 (المؤشر $(0 + \text{len}(\text{arr})) / 2$).
 - بما أن x يساوي العنصر الأوسط ، فقم بإرجاع مؤشر العنصر الأوسط ، وهو 2.
 - لذلك ، سترجع خوارزمية البحث الثنائي القيمة 2.
- البحث الثنائي هو خوارزمية فعالة للغاية للبحث في البيانات المرتبة. أسوأ حالة وقت يُعد تعقيد البحث الثنائي $O(\log n)$ ، حيث n هو عدد العناصر في المصفوفة. هذا يعني أن الوقت الذي يستغرقه البحث في المصفوفة ينمو لوغاريتمياً مع عدد العناصر ، وهو أفضل بكثير من تعقيد الوقت الخطي لخوارزمية بحث بسيطة.
- البحث الثنائي هو خوارزمية أساسية في علوم الكمبيوتر ويستخدم في مجموعة متنوعة من التطبيقات ، مثل البحث عن الملفات في نظام الملفات ، والعثور على السجلات في قاعدة البيانات ، و فرز البيانات