

جستجو

یک الگوریتم جستجو به طور کلی الگوریتمی است که درون یک مجموعه از داده ها که توسط یک نوع ساختمان داده ذخیره شده اند مکان یک مقدار داده شده به عنوان آرگومان جستجو را درون ساختمان داده مشخص می کند، یا تعیین می کند در مجموعه وجود دارد یا خیر.

[جستجوی خطی](#)

[جستجوی دودویی](#)

جستجوی خطی

جستجوی خطی (linear search) یا جستجوی ترتیبی (sequential search) کلیه عناصر درون یک لیست را یکی یکی بررسی می کند تا آرگومان جستجو پیدا شود.

شبه کد زیر روش جستجوی خطی را نشان می دهد. مقدار x درون آرایه A با n عنصر جستجو می شود و موقعیت آنرا بر می گرداند. اگر در آرایه وجود نداشته باشد صفر برگردانده می شود.

```
i:=0;
for i:=1 to n do
  if A[i] = x then
    break
  end if
end for
Return i
```

اگر تعداد عناصر مجموعه n باشد، زمان جستجو $O(n)$ است. بهترین حالت زمانی اتفاق می افتد که آرگومان جستجو برابر با اولین عنصر لیست باشد که با یک مقایسه پیدا می شود. بدترین حالت وقتی است که داده درون لیست وجود ندارد یا در انتهای لیست واقع شده است که n مقایسه مورد نیاز است.

اگر تعداد عناصر کم باشد جستجوی خطی به دلیل سادگی از الگوریتم های پیچیده دیگر مناسب تر است. برای لیست های نامرتب اغلب جستجوی خطی اولین انتخاب است. کارایی الگوریتم روی یک لیست مرتب بالا می رود. در این حالت به جای رسیدن به انتهای لیست، جستجو با رسیدن به اولین عنصری که بزرگتر (یا کوچکتر) از آرگومان جستجو است خاتمه پیدا می کند.

جستجوی دودویی

الگوریتم جستجوی دودویی (binary search algorithm) روشی برای جستجوی یک مقدار درون یک لیست مرتب است. عنصر وسط لیست انتخاب شده و با آرگومان جستجو مقایسه می شود تا تعیین شود از آن بزرگتر، کوچکتر یا مساوی است. اگر آرگومان از عنصر انتخاب شده بزرگتر باشد جستجو در نیمه پایینی و اگر کوچکتر باشد در نیمه بالایی لیست ادامه پیدا می کند.

کد بازگشتی جستجوی دودویی به صورت زیر است:

```
int BinarySearch(int A[], int value, int low, int high) {
  if (high < low)
    return -1 // not found
  mid = (low + high) / 2
  if (A[mid] > value)
    return BinarySearch(A, value, low, mid-1)
  else if (A[mid] < value)
    return BinarySearch(A, value, mid+1, high)
  else
    return mid // found
}
```

زمان جستجو $O(\log n)$ است که زمان بهتری نسبت به جستجوی خطی است. اگر آرگومان جستجو برابر با عنصر وسط لیست باشد با یک مقایسه پیدا می شود که بهترین حالت است. در بدترین حالت به $\lfloor \log_2^n \rfloor + 1$ مقایسه نیاز است. جستجوی دودویی مثالی از یک الگوریتم تقسیم و غلبه است.