

الگوریتم جستجو فقط برای آرایه می باشد؟!

خیر ، بسته به نوع مسئله می توان از الگوریتم های جستجوی گوناگون استفاده کرد . آرایه ها و لیست ها از نوع داده های خطی هستند ، بنابراین می توان از الگوریتم های مشابهی برای جستجو درون آنها استفاده کرد . ولی برای ساختارهای غیر خطی مانند گراف و درخت از الگوریتم های دیگری استفاده می کنیم . برای مثال برای جستجو درون درخت های ساده دو الگوریتم زیر از ساده ترین و پرکاربردترین الگوریتم ها هستند :

- جستجوی اول سطح (Breath First Search / BFS)
- جستجوی اول عمق (Depth First Search / DFS)

<https://fa.wikipedia.org/wiki/%D%BE%DB%AC%D%85%D%A%AV%DB%AC%D%84%BA%AF%D%B%D%A%AV%D%81>

<https://hackernoon.com/graphs-in-cs-and-its-traversal-algorithms-cfee5533f74e>

آیا الگوریتم جستجو فقط برای آرایه اعداد است ، برای string و object کار نمی کند؟!

آری ، البته این موضوع بستگی به زبان و زمینه ی مورد استفاده دارد . برای مثال در مورد اشیاء باید ابتدا مشخص شود که نحوه ی مقایسه ی آنها در زبان برنامه نویسی مورد نظر چگونه است ، بعد در مورد اینکه چگونه می توان آنها را در یک لیست یا مجموعه قرار داد صحبت کرد و بعد به سراغ جستجو در مورد آنها رفت . برای مثال در زبان سی شارپ با پیاده سازی اینترفیس IComparable می توان نحوه ی مقایسه ی یک کلاس و اشیایی که از آن نوع کلاس ساخته می شوند را تعیین کرد و بر اساس آن لیستی از اشیاء ساخت و بعد جستجوی خطی یا باینری را بر روی آن اعمال کرد .

<https://stackoverflow.com/questions/21961859/c-sharp-list-of-objects-binary-search-with-a-comparer>

<https://coding.abel.nu/2014/09/net-and-equals/>

آیا الگوریتمی برای جستجوی گراف وجود دارد؟!

وجود دارد. تقریباً تمامی برنامه های نقشه خوان و مسیریاب (مانند Waze ، بلد ، نشان و ...) براساس الگوریتم های جستجوی گراف کار می کنند .

در اینجا بعضی از این الگوریتم ها را می

- <http://www.sourcecodes.ir/post.php?id=۳۲۷&title=-گراف-های-ییمایش-گراف>

خیر، پیش نیاز این الگوریتم مرتب بودن آرایه است.

- مرتب سازی حبابی
- مرتب سازی انتخابی
- مرتب سازی درجی
- و ...

<https://fa.wikipedia.org/wiki/%D8%A7%D9%84%DA%AF%D9%88%D8%B1%DB%8C%D8%AA%D9%A5%0F%09%A5%0A%01%0A%00%0A%0A%02%A%0A%03%0A%04%0A%02%00%A0%09%A1%09%A4%0A%01%0A%03%0A%00%0A%04%09%A4%00%00%09%A4%0A%01%00%A0%0A%00%09%A5%02%A%0A%09%A4%0A%04%00%A0%09%A5%0A%01%0A%00%0A%0A%02%A%0A%03%0A%04%0A%02%00%A0>

الگوریتم جستجوی درون یابی بر روی لیست یا آرایه مرتب به کار می رود که توزیع داده در آنها تقریباً یکسان باشد. جستجوی درون یابی با توجه به مقدار کلیدی که در حال جستجوی آن می باشد به نقاط متفاوتی مراجعه میکند. یعنی اگر به فرض مقدار کلید به آخرین عنصر نزدیکتر باشد جستجوی درون یابی، جستجو را از سمت آخر شروع میکند.

برای پیدا کردن موقعیت مورد نظر از فرمول زیر استفاده می کنیم .

$$mid = low + \lfloor \frac{x - S[low]}{S[high] - S[low]} \times (high - low) \rfloor$$

برای مثال، اگر مقدار اندیس یک آرایه ی S برابر ۴ و مقدار اندیس ۱۰ برابر ۹۷ باشد و ما به دنبال $x=25$ باشیم، محاسبه زیر را انجام می دهیم :

$$mid = 1 + \lfloor \frac{25 - 4}{97 - 4} \times (10 - 1) \rfloor = 1 + \lfloor 2.032 \rfloor = 3$$

جستجوی درون یابی نوع تغییر شکل یافته از جستجوی دو دویی است .

پیچیدگی زمانی این الگوریتم در زمانی که داده های با نسبت یکسان توزیع شده باشند برابر $\log(\log(n))$ ، در بدترین حالت برابر $O(N)$ و در بهترین حالت برابر $O(1)$ می باشد

https://fa.wikipedia.org/wiki/%D8%AC%D8%B3%D8%AA%D8%AC%D9%88%DB%8C_%D8%AF%D8%B1%D9%88%D9%86%E2%80%8C%DB%8C%D8%A7%D8%A8%DB%8C