Axtarış alqoritmləri elementi yoxlamaq və ya onun saxlandığı hər hansı bir məlumat strukturundan elementi əldə etmək üçün nəzərdə tutulmuşdur.

əməliyyatının növündən asılı olaraq bu alqoritmlər ümumiyyətlə iki kateqoriyaya bölünür:

1. **Ardıcıl Axtarış** (**Sequential Search**): Burada siyahı və ya massiv ardıcıl olaraq keçilir və hər bir element

yoxlanılır. Məsələn: [**Xətti Axtarış**](https://www.geeksforgeeks.org/linear-search/).

1. **Interval Axtarış** (**Interval Search**): Bu alqoritmlər xüsusi olaraq çeşidlənmiş məlumat strukturlarında axtarış üçün nəzərdə tutulmuşdur. Bu tip axtarış alqoritmləri Xətti Axtarışdan qat-qat səmərəlidir, çünki onlar dəfələrlə axtarış strukturunun mərkəzini hədəf alır və axtarış sahəsini yarıya bölürlər. Məsələn: [**İkili Axtarış**](https://www.geeksforgeeks.org/binary-search/)**.**

**Xətti axtarış** (en:linear search) indexlere bir bir baxir massivdəki məxsusi elementi tapmaq üçün massivdəki hər bir elementi ardıcıl şəkildə yoxlayır və bu elementi tapana qədər davam edir.(çünki verilənlər qarışıq şəkildədir ardıcıl deyil) Bu alqoritmin vaxt mürəkkəbliyi O(n)-dir

**İkili Axtarış** :axtarış intervalını dəfələrlə yarıya bölməklə çeşidlənmiş massivdə istifadə edilən axtarış alqoritmidir. Binar axtarışın ideyası massivin çeşidlənməsi haqqında məlumatdan istifadə etmək və vaxt mürəkkəbliyini O(Log n)-ə endirməkdir.

**Jump axtarış** çeşidlənmiş massivlər üçün axtarış alqoritmidir. Konsepsiya, hədəfin daxilində ola biləcəyi elementlər blokunu tapmaq üçün əvvəlcədən müəyyən edilmiş məbləğlə irəli atılmaqla xətti axtarışdan daha az element axtarmaqdır. Hədəfin olduğu bloku tapdıqdan sonra hədəfi axtarmaq üçün xətti axtarış aparacaq.

Jump Search O(√ n) zaman mürəkkəbliyini yaradır.Jump Search-in vaxt mürəkkəbliyi Xətti Axtarış O(n) və İkili Axtarış O (Log n) arasındadır.İkili Axtarış Jump Axtarışdan daha yaxşıdır, lakin Jump axtarışının bir üstünlüyü var ki, biz geriyə yalnız bir dəfə gedirik

**İnterpolyasiya axtarışı**- axtarılan açarın dəyərinə görə müxtəlif yerlərə gedə bilər.Məsələn, açarın dəyəri sonuncu elementə yaxındırsa, interpolyasiya axtarışı çox güman ki, axtarışa son tərəfə doğru başlayacaq.

Binar Axtarış üzərində təkmilləşdirmədir. İkili Axtarış həmişə yoxlamaq üçün orta elementə keçir. İnterpolyasiya Axtarış vaxtının mürəkkəbliyinin artım tempi Binary Search ilə müqayisədə daha kiçikdir. İnterpolyasiya Axtarışı üçün ən yaxşı hal orta (bizim yaxınlaşma) istədiyiniz açar olduqda baş verir. Bu, ən yaxşı zaman mürəkkəbliyini O(1) edir.

**Eksponensial axtarış** iki addımı əhatə edir: Zaman mürəkkəbliyi O(Log n)

1.Elementin mövcud olduğu aralığı tapın

2.Yuxarıda tapılmış diapazonda Binar Axtarış edin.

**Alt siyahı** (sublist)axtarışı birinci siyahının 2-ci siyahıda olub-olmadığını yoxlamaqdır. İki dəyər uyğun gəlmirsə, o, ikinci siyahının növbəti elementinə keçir.

Zamanın mürəkkəbliyi: O(m\*n) burada m ikinci siyahıdakı qovşaqların sayıdır və n-də birinci siyahıda

**Fibanocci**- array və orada axtarılacaq x elementi verilmişdir.

X-in indeksini qaytarın, əgər o massivdə varsa, əks halda -1 qaytarın.

O(Log n) zaman mürəkkəbliyinə malikdir