**Интерполяционный поиск**

Речь идёт о поиске позиции объекта (числа, строки) в упорядоченной (по возрастанию, убыванию) коллекции (массив, список). Алгоритм интерполяционного поиска - это доработка алгоритма бинарного поиска. В бинарном интервал делится на две равные части, выбирается элемент в середине коллекции и сравнивается с искомым.

Решающее правило алгоритма бинарного поиска:

1) если искомый равен элементу в середине списка, то позиция найдена;

2) если искомый элемент больше, то левая граница поиска сдвигается к середине (плюс 1);

3) если искомый элемент меньше, то правая граница поиска сдвигается к середине (минус 1).

Асимптотика сложности алгоритма бинарного поиска равна *log2(N)*, то есть для 1024 элементов количество шагов поиска в худшем случае будет N+1.

Пример бинарного поиска значения 5 в массиве длиной 19:



Теперь рассмотрим интерполяцию.

Интерполяция — это способ нахождения промежуточных значений величины по имеющемуся дискретному набору известных значений. В нашем случае дискретный набор значений хранится в коллекции. Чем более равномерно там размещены значения, тем более эффективен будет интерполяционный поиск. Интерполяционный поиск эффективен на коллекциях значительного объёма. Асимптотика сложности интерполяционного поиска log2( log2(N) ).

Рассмотри алгоритм интерполяционного поиска.

Основное его отличие от бинарного в способе поиска элемента, делящего весть диапазон на две части. В бинарном — примерно посередине. В интерполяционном — в пропорции искомого значения к диапазону индексов элементов.

Пример интерполяционного поиска значения 11 в массиве длиной 9:

ПЕРВЫЙ ШАГ:

left = 0; right = 8

shift = (findValue - leftValue) / (rightValue - leftValue) = (11 - 1) / (14 - 1)

len = right - left = 8 — 0 = 8

mid = left + shift \* count = 0 + 10/13 \* 8 = 6,15 = 6

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 3 | 4 | 6 | 9 | 11 | 12 | 13 | 14 |

ВТОРОЙ ШАГ:

left = 0; right = 5

shift = (findValue - leftValue) / (rightValue - leftValue) = (11 - 1) / (11 - 1)

len = right - left + 1 = 5 - 0 + 1

mid = left + shift \* count = 10/13 \* 9 = 6

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | 3 | 4 | 6 | 9 | 11 | 12 | 13 | 14 |



При пошаговом выполнении интерполяционного поиска индекса значения 5 в массиве длиной 19:

- на первом шаге: левая граница = 0, правая = 19;

вычисляем mid = левая граница + смещение

где смещение = отношение разницы между **значением искомого элемента и значением элемента на левой границе** поделённой на разницу между индексами искомого элемента и индекса левой границы

mid = () / ()

- на втором шаге: левая граница = 11, правая = 19;