|  |  |
| --- | --- |
| **26** | 7,8,12,16,19,21,21,27,34,34,39,40,42,46,48,49,53,54,57,59,70 |

1. **Интепполяционный поиск**

Шаг 1. Рассчитаем расстояние шага (d) по формуле

d=

i=1; j=21; K=53; Ki=7; Kj=70

d= = 14

Сравниваем ключ, стоящий под номером 15, с отыскиваемым ключом:

K15 ∨ K → 48 < 53 → сужается область поиска

**48,49,53,54,57,59,70**

Щаг 2.

i=1; j=7; K=53; Ki=48; Kj=70;

d= = 1

K2 ∨ K → 49 < 53 → сужается область поиска

**49,53,54,57,59,70**

Щаг 3.

i=1; j=6; K=53; Ki=49; Kj=70

d= = 0

Следовательно, алгоритм заканчивает работу. Воспользуемся линейным (последовательным) поиском.

Шаг 4. Установим i=1, k1=49, k1≠K.

Шаг5. Увеличим i на 1, k2=53, k2=K. Ключ найден.

Искомый ключ удалось найти за 5 шагов. Временная сложность алгоритма О(5).



1. **Поиск по бинарному дереву**

7,8,12,16,19,21,21,27,34,34,39,40,42,46,48,49,53,54,57,59,70

N=[21/2]+1=11. 11 элемент — это ключ равный 39. Построим дерево так, чтобы оно было сбалансированным.



Ключ удалось найти за 2 шага. Временная сложность алгоритма О(2).



1. **Поиск хешированием**

7,8,12,16,19,21,21,27,34,34,39,40,42,46,48,49,53,54,57,59,70

m = Kmax/2=35

h(k) = k mod (m)

h(k) ={7, 8, 12, 16, 19, 21, 21, 27, 34, 34, 4, 5, 7, 11, 13, 14, 18, 19, 22, 24, 0}

h(k) =53 mod (35) = 18

Это значит, что ключ К=53 может быть только в строке со значением хэш-функции 18. Коллизии отсутствуют. Ключ был найден за 1 шаг. Временная сложность алгоритма О(1).



**Вывод:** Наименьшее кол-во шагов, за которое можно было найти ключ, было достигнуто при поиске хэшированием (1 шаг). Следующее наименьшее количество шагов было достигнуто при поиске с помощью бинарного дерева (2 шага). Наибольшее количество шагов (5 шага) было при интерполяционном поиске, поэтому его эффективность наименьшая среди остальных методов поиска.