SEARCHING ALGORITHMS

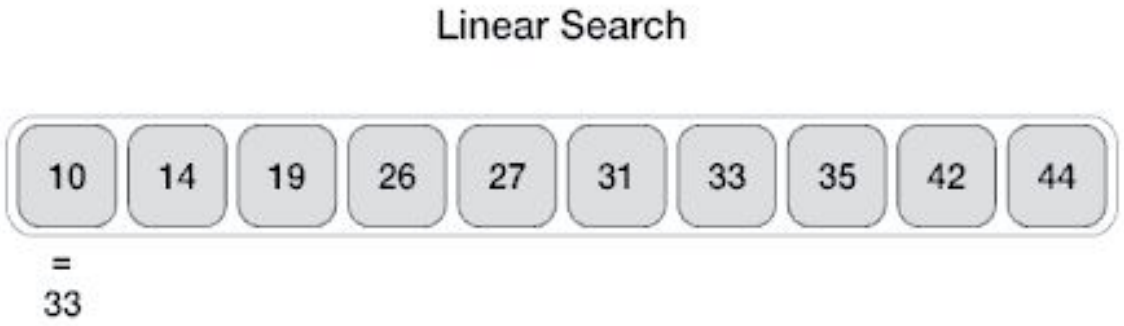
Алгоритми за търсене

- Linear search
- Jump search
- Binary search
- Ternary Search
- Interpolation search
- Exponential search
- Fibonacci search

Linear search

- Идея - Итерираме последователно докато не намерим търсената от нас стойност.

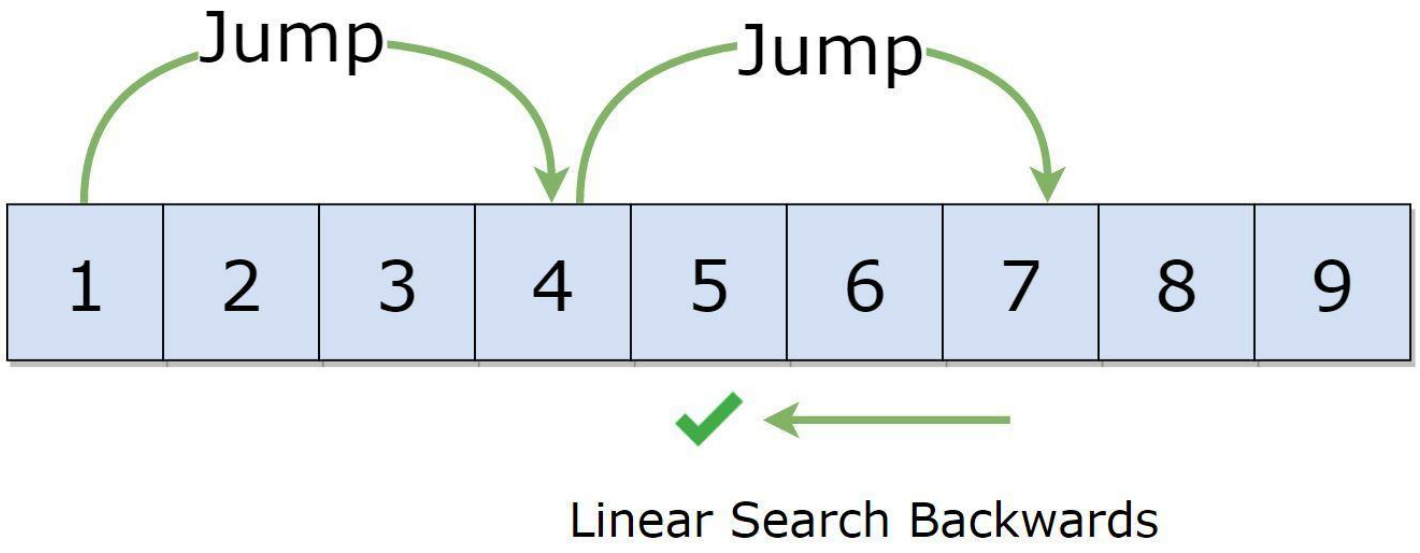
Best-case performance	*O(1)*
Average-case performance	*O(n)*
Worst-case performance	*O(n)*



Jump Search

- Идея - Прескачаме по √n елемента докато не достигнем елемент, по-голям от търсения.
- Връщаме се назад, √n позиции или докато не открием търсеният от нас елемент.

Best-case performance	*O(1)*
Average-case performance	*O(√n)*
Worst-case performance	*O(√n)*



Binary search

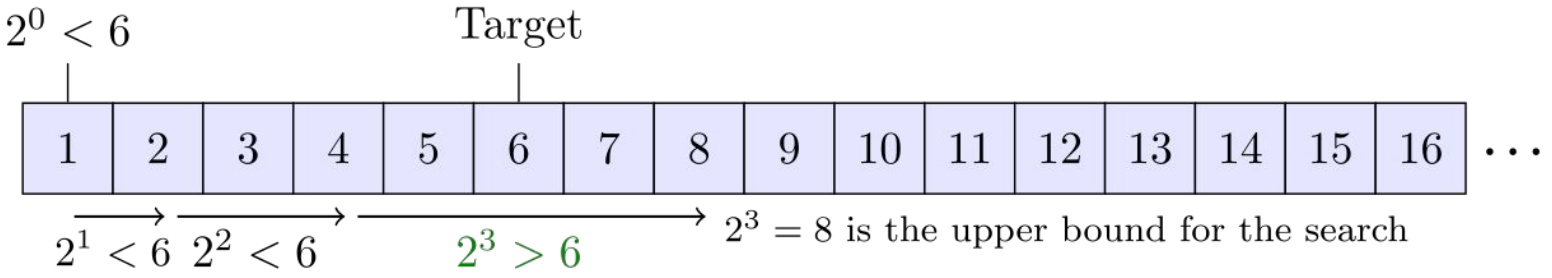
- Прилага се върху сортирани данни или върху краен интервал от стойности [left, right].
- Идея:
  - Фиксираме средата на интервала mid (mid = (left + right) / 2).
  - Ако сме открили търсената от нас стойност или интервалът е празен, спираме процедурата.
  - Ако търсената от нас стойност е по-голяма от фиксираната, продължаваме процедурата в десният интервал, породен от разделянето (mid + 1, right).
  - В противен случай в левият (left, mid - 1).

Best-case performance	*O(1)*
Average-case performance	*O(log n)*
Worst-case performance	*O(log n)*

Exponential Search

Best-case performance	*O(1)*
Average-case performance	*O(log n)*
Worst-case performance	*O(log n)*

- Идея - Проверяваме елементите на позиции 2^0, 2^1, ..., 2^i, ... докато елемента на позиция 2^i е по-малък или равен на търсения.
- Така за търсения от нас елемент x е вярно: $A[2^{i-1}] < x <= A[2^i]$.
- Прилагаме двоично търсене върху елементите от интервала: $[2^{i-1} + 1 ... 2^i]$.



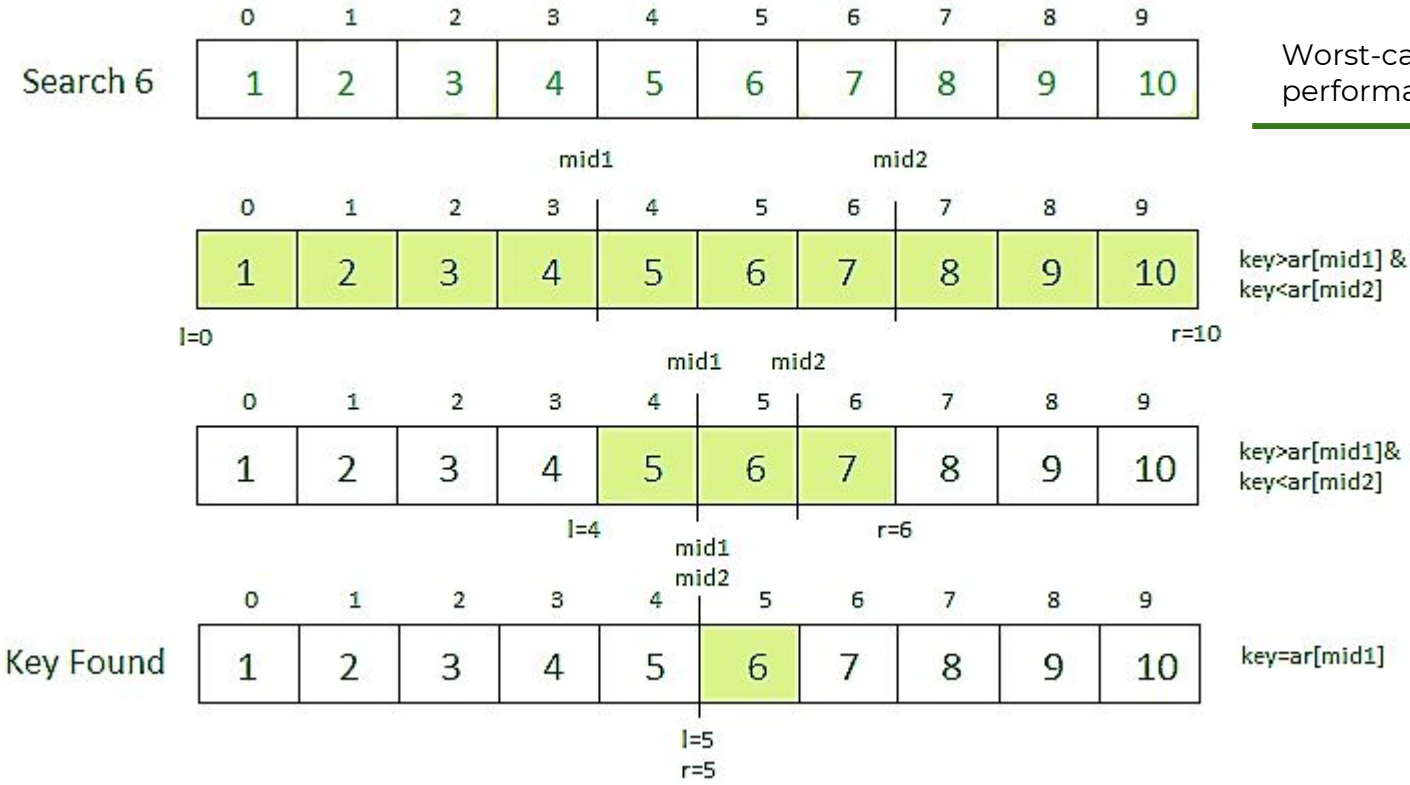
Ternary Search

- Прилага се върху сортирани данни или върху краен интервал от стойности [left, right].
- Идея:
  - Разделяме интервала на 3 “равни” части: [left, m1], [m1, m2], [m2, right].
  - Ако сме открили търсената от нас стойност или интервалът е празен, спираме процедурата.
  - Продължаваме процедурата в интервалът, на когото принадлежи търсената от нас стойност.

Best-case performance	*O(1)*
Average-case performance	*O(log n)*
Worst-case performance	*O(log n)*

Ternary Search

Best-case performance	*O(1)*
Average-case performance	*O(log n)*
Worst-case performance	*O(log n)*



Задачи

- Намерете $\sqrt[3]{n}$ с точност до третия знак след десетичната запетая, където n се въвежда от клавиатурата.
- Sysadmin
- Sequence
- Exam
- Load
- Increase
- Socks
- Shoe Shopping
- Article
- Ферма
- Riddles
- Motherboard
- Graze


THE
END