

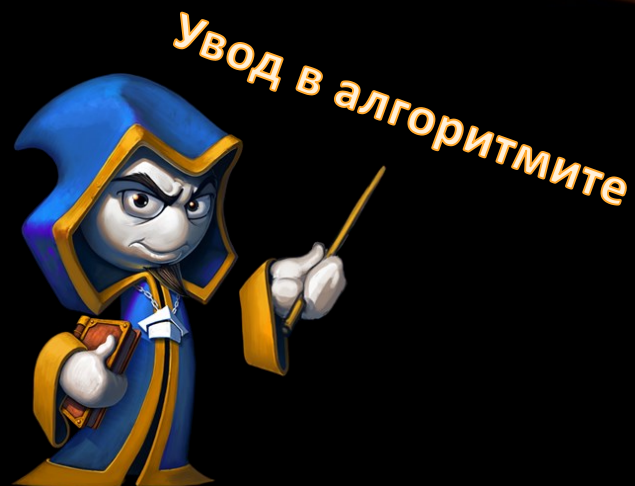
Алгоритми за търсене



Учителски екип

Обучение за ИТ кариера

<https://it-kariera.mon.bg/e-learning/>



Съдържание

1. Типове търсене и тяхното приложение
2. Последователно търсене
3. Двоично търсене
4. Търсене чрез интерполация



Алгоритъм за търсене

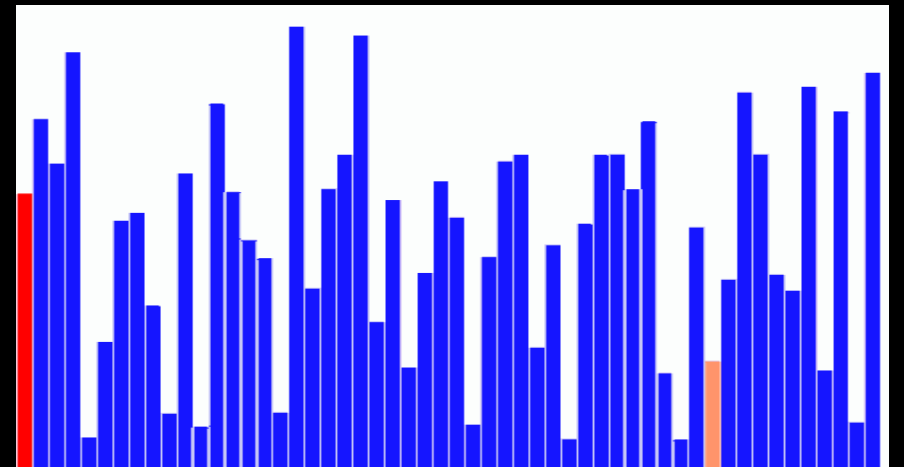
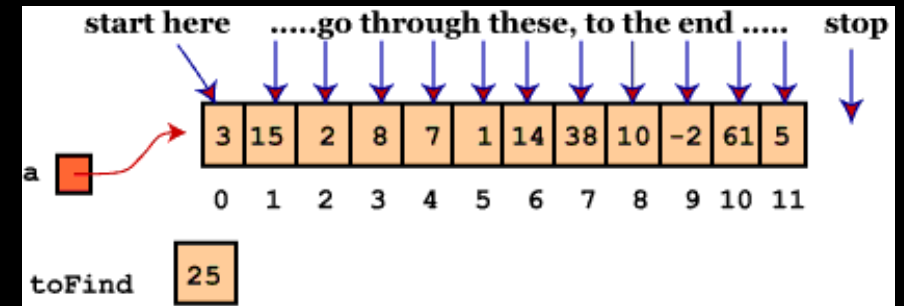
- **Алгоритъм за търсене** == алгоритъм за намиране на елемент с указани свойства всред колекция от елементи
- Различни типове алгоритми за търсене
 - За виртуални пространства за търсене
 - Удовлетворяващи дадени математически равенства
 - Опит за ползване на частично знание за структура
 - За подструктура на дадена структура
 - Граф, низ, крайна група
 - Търсене на \min / \max на функция и т.н.



Последователно (линейно) търсене

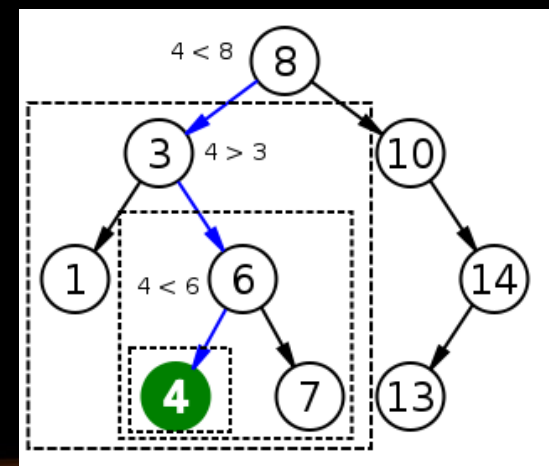
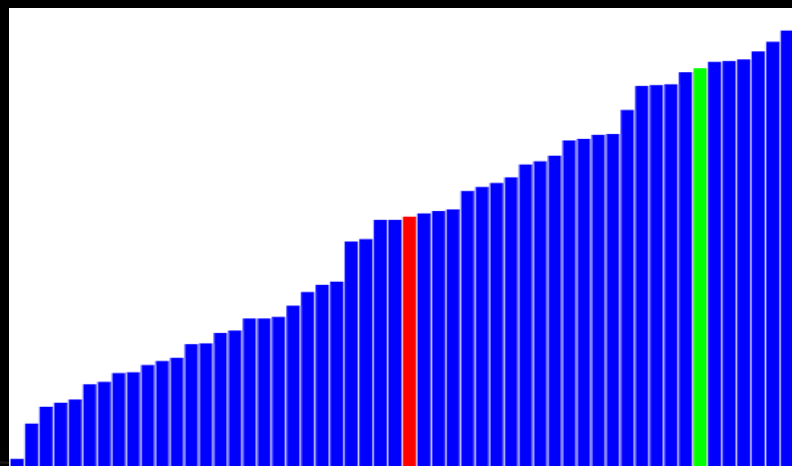
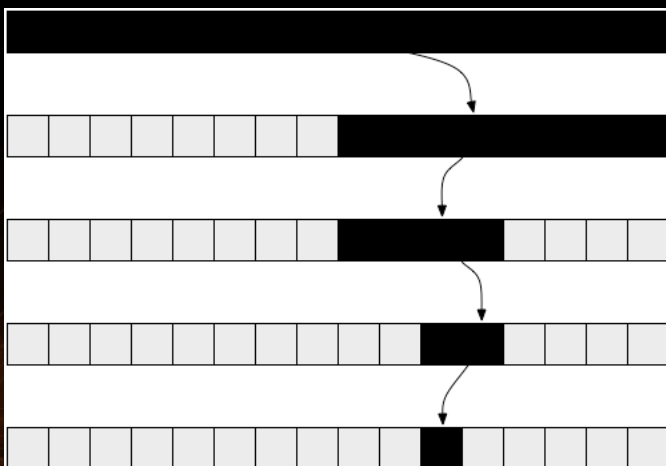
- Последователно (или линейно) търсене (linear search) намира определена стойност в списък (онагледяване)
 - Проверява всеки от елементите
 - Един по един, последователно
 - Докато открием желани
- Най-лошо и средно изпълнение: $O(n)$

```
for each item in the list:  
    if that item has the desired value,  
        return the item's location  
return nothing
```



Двоично търсене

- Двоичното търсене (binary search) намира елемент в подредена структура от данни
- На всяка стъпка, сравнява въведеното със средния елемент
 - Алгоритъмът продължава да търси в лявата или дясната подструктура
- Обичайно изпълнение: $O(\log(n))$
- Вижте онагледяването



Двоично търсене (рекурсивно)

```
int BinarySearch(int arr[], int key, int start, int end) {  
    if (end < start)  
        return KEY_NOT_FOUND;  
    else {  
        int mid = (start + end) / 2;  
        if (arr[mid] > key)  
            return BinarySearch(arr, key, start, mid - 1);  
        else if (arr[mid] < key)  
            return BinarySearch(arr, key, mid + 1, end);  
        else  
            return mid;  
    }  
}
```

Двоично търсене (с итерация)

```
int BinarySearch(int arr[], int key, int start, int end) {  
    while (end >= start) {  
        int mid = (start + end) / 2;  
        if (arr[mid] < key)  
            start = mid + 1;  
        else if (arr[mid] > key)  
            end = mid - 1;  
        else  
            return mid;  
    }  
    return KEY_NOT_FOUND;  
}
```

Търсене чрез интерполация

- Търсене чрез интерполация (Interpolation search) == алгоритъм за търсене по даден ключ в подреден индексирани масив
 - Подобно на това как хората търсят в телефонен указател
 - Изчислява къде в оставащата част трябва да е търсения елемент
 - Двоичното търсене винаги избира средния елемент
 - Може да имаме по-добро попадение, например *Ангел* трябва да е в началото, а не в средата на телефонния указател, нали?
- Обичайният случай: $\log(\log(n))$, Най-лошия: $O(n)$
- http://youtube.com/watch?v=l1ed_bTv7Hw

Търсене чрез интерполация – примерна реализация

```
int InterpolationSearch(int[] sortedArray, int key) {  
    int low = 0;  
    int high = sortedArray.Length - 1;  
    while (sortedArray[low] <= key && sortedArray[high] >= key) {  
        int mid = low + ((key - sortedArray[low]) * (high - low))  
            / (sortedArray[high] - sortedArray[low]);  
        if (sortedArray[mid] < key)  
            low = mid + 1;  
        else if (sortedArray[mid] > key)  
            high = mid - 1;  
        else  
            return mid;  
    }  
    if (sortedArray[low] == key) return low;  
    else return KEY_NOT_FOUND;  
}
```

Обобщение

- Алгоритмите за търсене намират широко приложение
- Последователното търсене се използва за не подредени списъци
- Двоичното търсене е за сортирани списъци, на всяка стъпка изключваме половината елементи
- Търсенето чрез интерполация е дори още по-бързо, изчисляваме къде е елемента



Алгоритми за търсене



Въпроси?



Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист"



Министерство
на образованието
и науката



Национална
програма
„Обучение за
ИТ кариера“

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под свободен лиценз **CC-BY-NC-SA**



SoftUni
Foundation

