Searching algorithms

By: Yehia M. Abu Eita

Outlines

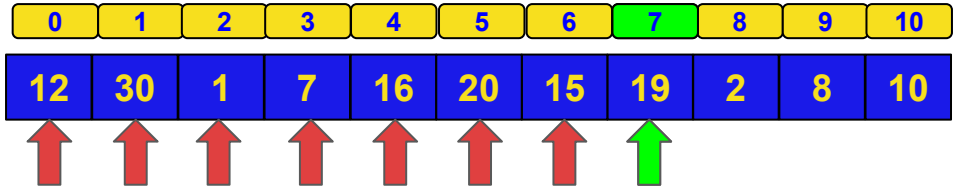
- Introduction
- Linear search algorithm
- Binary search algorithm
- Jump search algorithm

Introduction

- Searching algorithms help in any searching functionality you need.
- These algorithms are used in **search engines**, search in **websites**, **eCommerce applications,** ...etc.
- Time complexity is very important for searching algorithms.
- **Less time is the best**.
- There are many **types** of searching algorithms:
  - **Linear search**
  - **Binary search**
  - **Jump search**

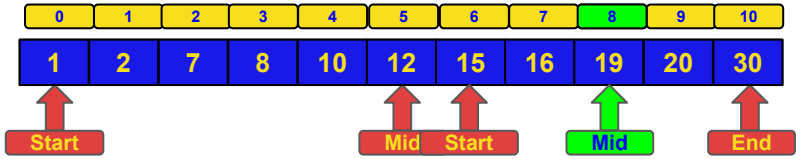
Linear search algorithm

- **Steps** to make linear search into an array:
  - **Check if array element is matched with the value you search for**
  - **If not matched, move to the next element**
  - **Repeat previous steps until you reach the last element, or find a matching element**
  - **If not matched and the last element, return error not found**
  - **If matched, return the index of that element**

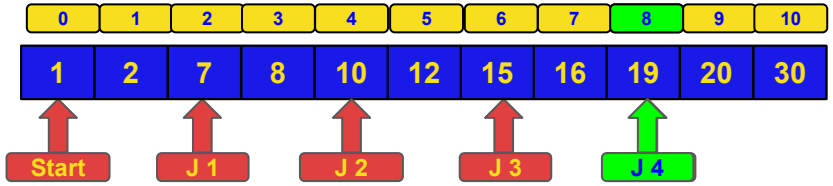


Binary search algorithm

- **Steps** to make binary search into an array:
  - Sort the array first
  - Calculate array mid, mid = (start + end) / 2
  - Check if the mid element is matched with the value you search for
  - If not matched, check if this element is larger than what you search for
  - If larger, change the end to be (mid - 1), then calculate a new mid
  - If not larger, change the start to be (mid + 1), then calculate a new mid
  - Repeat previous steps until you reach the last element, or find a matching element
  - If not matched and the last element (start = end = mid), return error not found
  - If matched, return the index of that element



Jump search algorithm



- **Steps** to make jump search into an array:
  - **Sort the array first**
  - **Calculate the jump, jump = min(sqrt(array_size))-1**
  - **Check if the first element, index 0, is matched with the value you search for**
  - **If not matched and smaller, add one jump to the index**
  - **If not matched and larger , return error not found**
  - **Check if the current element, index i, is matched with the value you search for**
  - **If not matched and smaller, add one jump to the index**
  - **If not matched and larger , make linear search backwards till the previous jump position**
  - **Repeat previous steps until you reach the last element, or find a matching element**
  - **If not matched and the last element, return error not found**
  - **If matched, return the index of that element**

Summary

- Now you are familiar with searching algorithms.
- Remember that linear search has O(n) time complexity.
- Remember that binary search has O(log n) time complexity.
- Remember that jump search has O(sqrt(n)) time complexity.