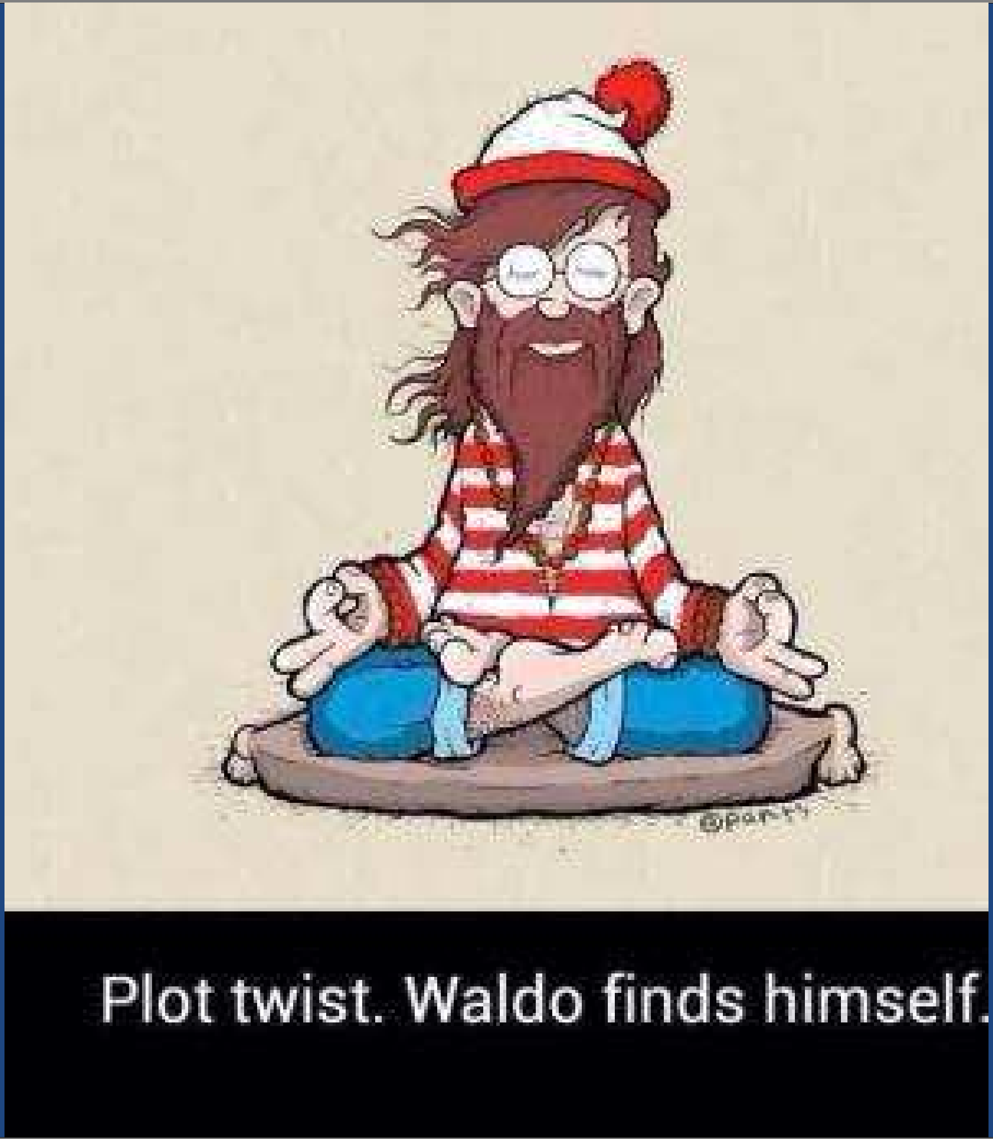
Slides #19

Searching



https://www.dailymoss.com/25-hilarious-wheres-waldo-jokes-will-make-rofl/

Topics

1) How can we search for an element in a vector or array?
   a) Linear Search – Just keep looking!
   b) Binary Search – I’m thinking of a number between 1 and 100....

Searching

- Searching involves...
  in a collection of items.
  - Ex: “Find the number <u>25</u> in the collection”
  - or sometimes: “Is the number <u>25</u> in the collection?”
  - and commonly: “Find <u>Bob</u>'s phone number.”
- Definitions:
  - Target element:
  - Search pool:

About searching

- There are many search algorithms.
  - Generally, we want the one which finds the element..
- A search can result in:
  - Finding the target element in the search pool (and returning its index), or
  - Proving that the target element is...

Linear search

- Linear search:
  -
    until have found the target element or have examined all elements.
- It's “linear” search because:
  - start with the first element and.. to the last element.

Linear search example

- Given the following search pool:
  Val: 8 19 71 5 16 27 38 40 0 56 26 10 24 30
- Use linear search to find the following (count comparisons):
  16 8 28

Linear search

```
// Find the index of the target element.
//	data:		Elements to search.
//	size:		Number of elements in data[]
//	target:		Value to find.
//	returns:	Index of target; -1 for not found.
int linearSearch (int data[], int size, int target)
{
	// Cycle through all elements
	for (int index = 0; index < size; index ++) {
		// When we find the item, return it's index.
		if (data[index] == target) {
			return index;
		}
	}
	// Item not found:
	return -1;
}
```

```
int main() {
	const int N = 5;
	int myData[] = {5, 10, 1, 18, 3};

	int pos = linearSearch(myData, N, 18);
	cout << "Index " << pos << endl;
	...
```

Binary search introduction

- Idea:
  - Each comparison...
- Similar to how to play "guess the number [1...100]".
  - Guess 50, it's less than that: [1 ... 49]
  - Guess 25, it's more than that: [26 ... 49]
  - Guess 37, it's less than that: [26 ... 36]
  - Guess 31, it's less than that: [26 ... 30]
  - Guess 28, it's more than that: [29 ... 30]
  - Guess 30, it's less that that: Answer is 29!
- Limitation:
  - Binary search works on...

Binary search description

- Binary search works as follows:
  - Start by looking at the middle element of the search pool.
    - If it's equal to the target, you are done!
    - If mid-element is less than the target...
    - If mid-element is greater than the target...
  - Repeat the above until:
    - You've found the element; or
    - There are...

Binary search example

Middle Formula:
(min + max) / 2

- Given the following search pool:

Idx:	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Val:	0	5	8	10	16	19	24	26	27	30	38	40	56	71

- Use binary search to find the following (count comparisons):

56 0 28

Binary search code

```
int binarySearch (int data[], int size, int target)
{
    int min=0, max=size-1, mid=0;
    // Narrow in the [min, max] bounds
    while (min <= max) {
        mid = (min+max) / 2;
        if (data[mid] == target) {
            return mid;
        } else {
            if (target < data[mid]) {
                max = mid-1;
            } else {
                min = mid+1;
            }
        }
    }
    return -1;   // Not found, return -1.
}
```

```
int main() {
    const int N = 5;
    int myData[] = {1, 3, 5, 10, 18};

    int pos = binarySearch(myData, N, 18);
    cout << "Index " << pos << endl;
    …
```

Linear vs binary search

- Comparisons: Which search ____?
  - Requires a sorted list:..
  - Slower (on average):..
  - Easier to understand, implement and debug...

- Algorithm Selection:
  - If it's easy to keep the data sorted or you'll be searching a lot, use binary search.
  - Otherwise, linear search may be better.

Review

- Fill in the following table for number of comparisons required to find elements in the following list.

2 5 7 8 11

	Linear Search	Binary Search
Find 7		
Find 11		
Find 6		

Summary

- Searching and Sorting are two classic computing science problems.
- Searching:
  - Linear: Look at each element to find item.
  - Binary: Look half way through sorted list to find which half target element could be in.