CSCI E-50 WEEK 3

TERESA LEE
SEPTEMBER 19, 2017

TODAY

- Recap: arrays & debug50
- asymptotic notation (O, Ω)
- linear search
- binary search
- bubble sort
- Selection sort
- recursion
- pset3

QUESTIONS?

ARRAY

To initialize an array

```
int score[0] = 0; // zero index all arrays!
int score[1] = 1;
int score[2] = 2;
```

or

```
int score[] = {0, 1, 2}; // size based on the number of entries
```

```
// make an array

<datatype> <name>[<size>];

    char alpha[26];

    Int score[5];

// iterate over the array's members
```

STRING

Just an array of characters!

Final index of a string in C is the null terminator ‘\0’, which tells a string that the string is over.

```
// declare string

String s = “teresa”;

// what happens when I index into s[i]?

Printf(“%c\n”, s[0]);

Printf(“%c\n”, s[1]);

Printf(“%c\n”, s[6]);

Printf(“%c\n”, s[7]);
```

Debug50

Example

Asymptotic Notation

- Algorithm’s running time
  - O (upper bound)
  - Ω (lower bound)
  - Θ upper and lower bounds are the same

Let's Look at an Example

Linear search:

What is the upper bound? O(n)

What is the lower bound? Ω(1)

Let's Look at an Example

Bubble Sort:

What is the lower bound? Ω(n)

Let’s Look at Example

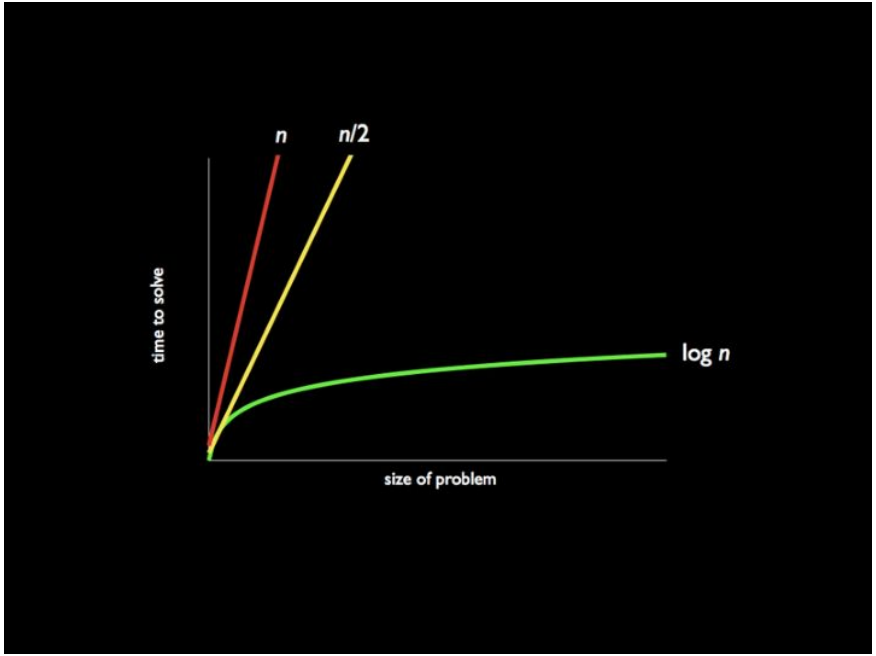
Bubble Sort:

What is the upper bound? O(n^2)

Running Time Summary

Algorithm	Big O	Big Ω
linear search	$O(n)$	$\Omega(1)$
binary search	$O(\log(n))$	$\Omega(1)$
bubble sort	$O(n^2)$	$\Omega(n)$
insertion sort	$O(n^2)$	$\Omega(n)$
selection sort	$O(n^2)$	$\Omega(n^2)$

Running Time Summary



Linear search

- O(n) , Ω(1)

```
// initialize an int array

Int haystack[] = {3, 2, 6}


// find the needle by using the
linear search
```

Let’s Look at an Example

Example 1 Linear Search

Binary search

- O(log n), Ω(1)
- Requirement - sorted list!
  - Why?

```
//pseudocode for binary search
```

Let’s Look at an Example

Example 2 Binary Search

Bubble Sort

- $O(n^2)$, $\Omega(n)$
- Pair-wise sorting
- Temporary variable?

Let’s Look at an Example

Example 3 Bubble Sort

Selection Sort

- $O(n^2)$, $\Omega(n^2)$
- Temporary variable?

```
// sort the list

Int list[] = {3, 13, 2, 86, 25}
```

Let’s Look at an Example

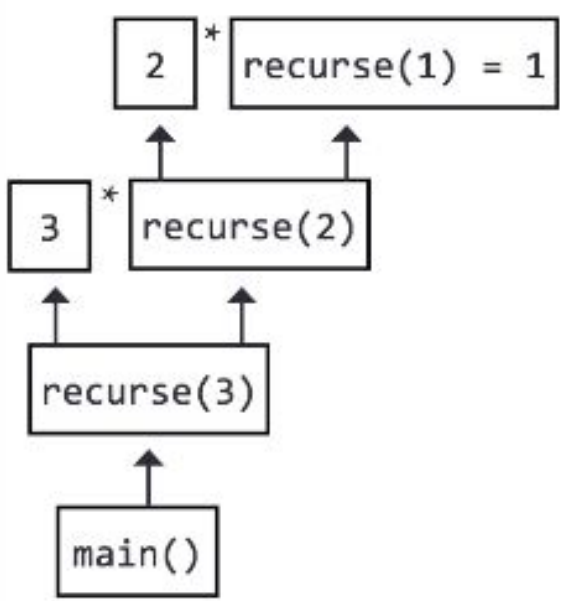
Example 4 Selection Sort

Recursion

- Recursive function calls itself as part of execution
- Cyclical use of a function
  - Every time you make a recursive call, there is a new stack frame
- You need:
  - Base case: when triggered, terminates the recursive process
  - Recursive case: where recursive process will actually occur

Let's Look at an Example

```
fact(1) = 1
fact(2) = 2 * fact(1)
fact(3) = 3 * fact(2)
fact(4) = 4 * fact(3)
fact(5) = 5 * fact(4)
...
```



pset3

Music

Shorts to Watch

- Computational Complexity
- Selection Sort
- Bubble Sort
- Insertion Sort
- Linear Search
- Binary Search
- Algorithms Summary
- Debugging
- Recursion

Final words on pset3

- Read background
- Read specification
- Watch Brian's walkthrough
- Remember to comment your codes!
  - Style Guide
- Test with check50!

ZOOM Office Hours

———

Tuesday 2pm - 5pm (Amrit)

Tuesday 9pm - 10pm (Pete)

Wednesday 8pm - 11pm (Pete, then Teresa)

Thursday 5pm - 6pm (Teresa)

Thursday 7pm - 9pm (Eric)

Friday 2pm - 4pm (Amrit)

Friday 7pm - 9pm (Pete)

Saturday 8pm - 10pm (Tom)