CS 105

Lecture Handout
Week 6: Oct. 5-7, 2015

Suppose you had nine “doors” with unknown numbers behind them:

??	??	??	??	??	??	??	??	??

Puzzle #1: If we know **nothing about the data**, what is the best (“optimal”) strategy for finding the number 42?

1) Start with first one, and go through each and every element

…we call this strategy:_**Linear Search**_____

Puzzle #2: If we know the **data is sorted**, what is the best (“optimal”) strategy for finding the number 42?

1) Pick a middle element

2) Remove the half of the data that is no longer relevant

3) Repeat 1-2 with the remaining data

…we call this strategy: _____**Binary Search**____________

Puzzle #3: In Computer Science, when we examine strategies for solving problems we often look at the **worst case running time** of an algorithm:

Amount of Data	Linear Search	Binary Search
7	7	3
15	15	4
31	31	5
n	n	$\ln(n+1)/\ln(2) = \log_2 n$
Key Takeaways:	**Advantage: Does not require the input data to be sorted. Easy to write and efficient for short lists. Disadvantage: very time consuming in case of long lists.**	**Advantage: Takes much lesser time Disadvantage: Requires the input data to be sorted.**

In Activity 5, you used the Illini Women’s Soccer Team data to understand arrays of objects:

```
var games = [
  { score: [4, 1],  opponent: "Oakland" },
  { score: [1, 0],  opponent: "Illinois State" },
  { score: [5, 2],  opponent: "TCU" },
  ...
];
```

Puzzle #4: Write a JavaScript function named `findOpponent` that uses a linear search to find an opponent in the `games` Array and return that game. (This function should return the entire object (eg: `games[i]`), not just the opponent’s name or score.)

```
var findOpponent = function(opponent, games) {
    for (var i=0; i< games.length; i++) {
        if (games[i].opponent == opponent) {
            return games[i];
        }
    }
};

var answer = findOpponent(“Purdue”, games);
alert(answer);
```

Puzzle #5: Given the Array `games` arranged in the same way as it was in the Activity and on the top of this page, can we use a **binary search** to search for an opponent?
No. Because it is not sorted.

Why?

If we cannot, what can we do to make it so?

Sorting

Suppose we want to sort our data:

50	**34**	**87**	**13**	**11**	**58**	**17**	**29**	**52**

There are a lot of ways to rearrange the data, let us agree on one:

1) Find the smallest value in the data

2) Swap the smallest value with the first element
 (now the first value is sorted)

3) Repeat (1)-(3) with the remaining data.

…we call this strategy: ________Selection Sort________.

Puzzle #6: Using our data set we want to sort, let's run our sorting algorithm:

Round #	50	34	87	13	11	58	17	29	52
1	11	34	87	13	50	58	17	29	52
2	11	13	87	34	50	58	17	29	52
3	11	13	17	34	50	58	87	29	52
4	11	13	17	29	50	58	87	34	52
5	11	13	17	29	34	58	87	50	52
6	11	13	17	29	34	50	87	58	52
7	11	13	17	29	34	50	52	58	87
8									

Puzzle #7: What is the **worst case running time** for each algorithm?

Data Size	Linear Search	Binary Search	Selection Sort
9	9	3-4	45 (=9+8+7+6 ----) ~ 81 (= 9*9 = 81)
100	100	6-7	10,000
n	n	log(n)	n^2

The following is the same data set as earlier:

```
var games = [
  { score: [4, 1],  opponent: "Oakland" },
  { score: [1, 0],  opponent: "Illinois State" },
  { score: [5, 2],  opponent: "TCU" },
 ...
];
```

Puzzle #8: Write a JavaScript function named **sortByOpponent** that uses a selection sort to sort the **games** array based on the name of the opponent.

```
var sortByOpponent = function(games) {

  // Loop through the array:

    for (var i=0; i<games.length; i++) {

    // Declare a variable to store the smallest element:
        var min = i;

    // Loop through the array again, looking for the smallest
    // element that has not been put in the correct position:

        for (var j=i+1; j<games.length; j++) {

            if(games[j].opponent < games[min].opponent) {
                min = j;
            }

        }
    // Swap the smallest element with the current element:

        var temp_opponent = games[i].opponent;
         games[i].opponent = games[min].opponent;
         games[min].opponent = temp_opponent;

     }

};
```

Reminder: CS 105 Midterm Exam
Monday, October 12, 2015, 7:30pm – 9:00pm
(Cannot make it? Last day to sign up for a conflict is Wednesday, Oct. 7th!)