

Puzzle #1:

Linear Search: Start from beginning, check every single element until find the target.

Puzzle #2:

Binary Search: Start with the middle of a sorted list.

Puzzle #3:

Worse case running time

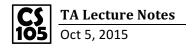
Amount of Data	Linear Search	Binary Search	
7	7	3	
15	15	4	
31	31	5	
n	n	log ₂ n	
Advantage	Data don't has to be sorted	Data has to be sorted	
Disadvantage	Less efficient	More efficient	

Puzzle #4:

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

Puzzle #7:

Data Size	Linear Search	Binary Search	Selection Sort
9	9	About 4	9*9 = 81
100	100	About 7	100*100 = 10,000
n	n	$\log_2 n$	n^2



Puzzle #8:

```
1 var sortByOpponent = function(games) {
3
    // Loop through the array:
 4
       for (var i=0;i<games.length;i++){</pre>
      // Declare a variable to store the smallest element:
 5
 6
           var min = i;
 7
      // Loop through the array again, looking for the smallest
 8
      // element that has not been put in the correct position:
9
           for (var j = i+1;j<games.length;j++){</pre>
10
11
               if (games[i].opponent<games[min].opponent){</pre>
12
                   min = i;
               }
13
14
           }
       // Swap the smallest element with the current element:
15
           var temp = games[i];
16
           games[i]=games[min];
17
           games[min]=temp;
19
       }
20 };
```