

1. Multiple choice/true/false
2. BubbleSort and SelectionSort both have $O(n^2)$ running time in every case - why is BubbleSort slower in practice?
3. Use Merge-Sort Tree to perform the MergeSort algorithm by hand on the array [8,7,6,5,4,3,2,1]. Show all steps.
4. Use the technique discussed in class to compute the running time of the following recursive algorithm.

```
function isEven(n){  
    if(n == 0)  
        return true;  
    if(n == 1)  
        return false;  
    return isEven(n-2);  
}
```

5. Devise an algorithm `secondSmallest(linkedlist)` in pseudo-code to return the secondsmallest element in the input linkedlist of integers. (You can use any linkedlist operations discussed in class.) Your algorithm must run in $O(n)$.
6. Devise an algorithm `elementsRangesIn(arr, x, y)` in pseudo-code to find the number of elements in `arr` which has value bigger than `x` and less than `y`. (`x` and `y` are two elements in the sorted array `arr`. `x` is less than `y`. And you can assume there are no duplicates in the array). Your algorithm must run in $O(\log n)$.

For example:

`elementsRangesIn ([1,5,6,7,8,9], 5, 8)` returns 2 since there are two elements (6 and 7) in between 5 and 8.