Homework Assignments #1

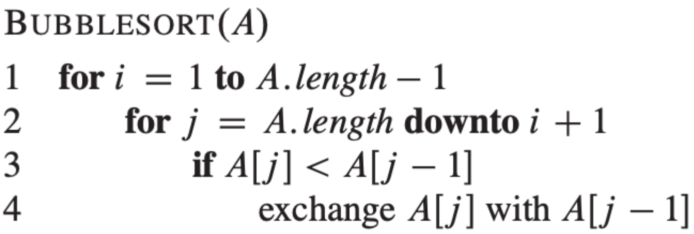
Due: 2020/04/08 12:00

Assessment policy:

- Give full points when correct, 1/n for solving each n subproblems. 0 for totally wrong or none, -1 for each error.
- There may be partial points for proofs if the direction is correct.

1. Correctness of bubble sort (11 pts)

Bubblesort is a popular, but inefficient, sorting algorithm. It works by repeatedly swapping adjacent elements that are out of order.



```
BUBBLESORT(A)
1  for $i = 1$ to $A.length - 1$
2      for $j = A.length$ downto $i + 1$
3          if $A[j] < A[j-1]$
4              exchange $A[j]$ with $A[j-1]$
```

To prove that `BUBBLESORT` is correct, we need to prove that it terminates and that $A'[1] \leq A'[2] \leq \cdots \leq A'[n]$, where $n = A.length$.

- (a) The loop invariant for the inner `for` loop (lines 2-4) is like follows. Prove that this loop invariant holds. *(3 pt)*

  > **loop invariant**: At the start of each iteration the subarray $A[j..n]$ consists of the elements originally in $A[j..n]$ before entering the loop but possibly in a different order and the first element $A[j]$ is the smallest among them.

- (b) Using the termination condition of the loop invariant proved in (a), state a loop invariant for the outer `for` loop (lines 1–4). Prove that this loop invariant holds. *(6 pt)*

- (c) What is the worst-case running time of bubblesort? How does it compare to the running time of insertion sort? *(2 pt)*

2. [Asymptotic growth rates] (10 pts)

List the sixteen functions below from asymptotically lowest order to highest order. If there are functions of the same order, indicate which.

n	2^n	$n \lg n$	$\ln n$
$n - n^3 + 7n^5$	$\lg n$	$\sqrt{n}$	e^n
$n^2 + \lg n$	n^2	2^{n-1}	$\lg \lg n$
n^3	$(\lg n)^2$	$n!$	$n^{1+\epsilon}$, where $0 < \epsilon < 1$

You can denote it as a subset notation, for instance like the follwing:

$O(n) \subset O(n^2) = O(3n^2 + 1) \subset O(n^3) \subset \cdots$

3. Brute-force maximum subarray problem (10 pts)

The maximum subarray problem takes $\Theta(n^2)$ time to compare all possible pairs.

- **input**: Sequence of n numbers A
- **output**: A subsequence A' whose sum is the maximum among all subsequences of A

Write a pseudocode for the brute-force method `BRUTE-FORCE-FIND-MAXIMUM-SUBARRAY(A)` of solving the maximum-subarray problem. Your procedure should run in $\Theta(n^2)$ time.

4. Apply substitution method (6 pts)

Prove the following solutions of recurrences with substitution method *(3 pts each)*

- (a) Show that the solution of $T(n) = T(n-1) + n$ is $O(n^2)$

- (b) Show that the solution of $T(n) = T(\lceil n/2 \rceil) + 1$ is $O(\lg n)$

5. Apply recursion method (15 pts)

Use a recursion tree to determine a good asymptotic upper bound for the following recurrences. Use the substitution methods to verify your answer. *(3 pts each)*

- (a) $T(n) = 3T(\lfloor n/2 \rfloor) + n$

- (b) $T(n) = T(n/2) + n^2$

- (c) $T(n) = 4T(n/2 + 2) + n$

- (d) $T(n) = 2T(n-1) + 1$

6. Draw recursion method (4 pts)

Draw the recursion tree for $T(n) = 4T(\lfloor n/2 \rfloor) + cn$, where c is a constant, and provide a tight asymptotic bound by the substitution method.