CPSC 5031 Algorithms HW #2 (5 pts)

Problems (all problems may be found in the Levitin textbook)**:**

Exercises 2.3 #6-a, b, c, d

6. Consider the following algorithm.

Algorithm *Enigma*$(A[0..n-1, 0..n-1])$
//Input: A matrix $A[0..n-1, 0..n-1]$ of real numbers
for $i \leftarrow 0$ **to** $n-2$ **do**
  for $j \leftarrow i+1$ **to** $n-1$ **do**
    if $A[i,j] \neq A[j,i]$
      return false
return true

a. What does this algorithm compute?
The algorithm determines if a matrix is symmetrical (mirror image I think is the term) on either side of its diagonal axis (top left to bottom right)

b. What is its basic operation?
The basic operation is checking for equality in the if statement

c. How many times is the basic operation executed?
n(n - 1) / 2 times in the worst case. In the best case, once.

d. What is the efficiency class of this algorithm?
O(n^2)

Exercises 2.4 #3

3. Consider the following recursive algorithm for computing the sum of the first n cubes: $S(n) = 1^3 + 2^3 + \cdots + n^3$.

Algorithm $S(n)$
//Input: A positive integer n
//Output: The sum of the first n cubes
if $n = 1$ **return** 1
else return $S(n-1) + n * n * n$

a. Set up and solve a recurrence relation for the number of times the algorithm's basic operation is executed. 2n

b. How does this algorithm compare with the straightforward nonrecursive algorithm for computing this function?

Submission

- Deadline: Thurs, 4/15/2021, 6:00pm
- Submit your solutions as a Word/PDF on Canvas