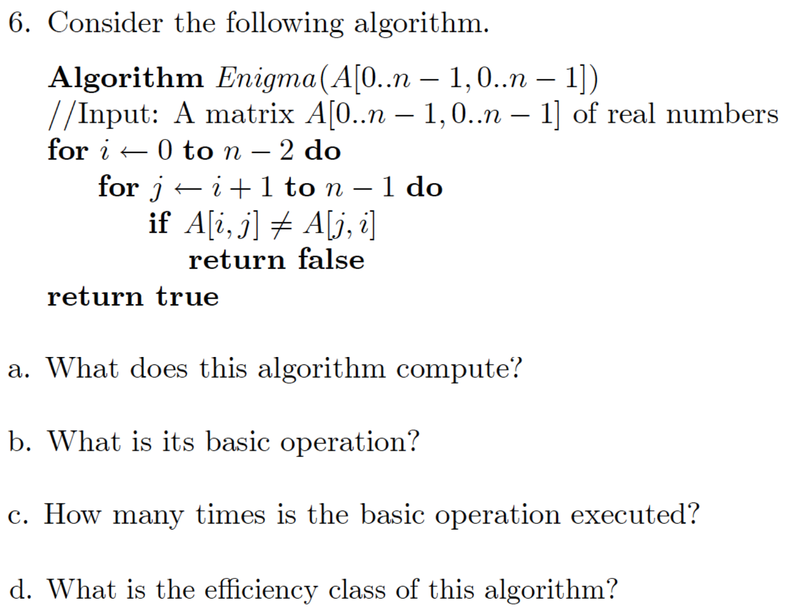
CPSC 5031 Algorithms
HW #2 (20 pts)

Exercises 2.3 #6-a,b,c,d (4 points)



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**
```

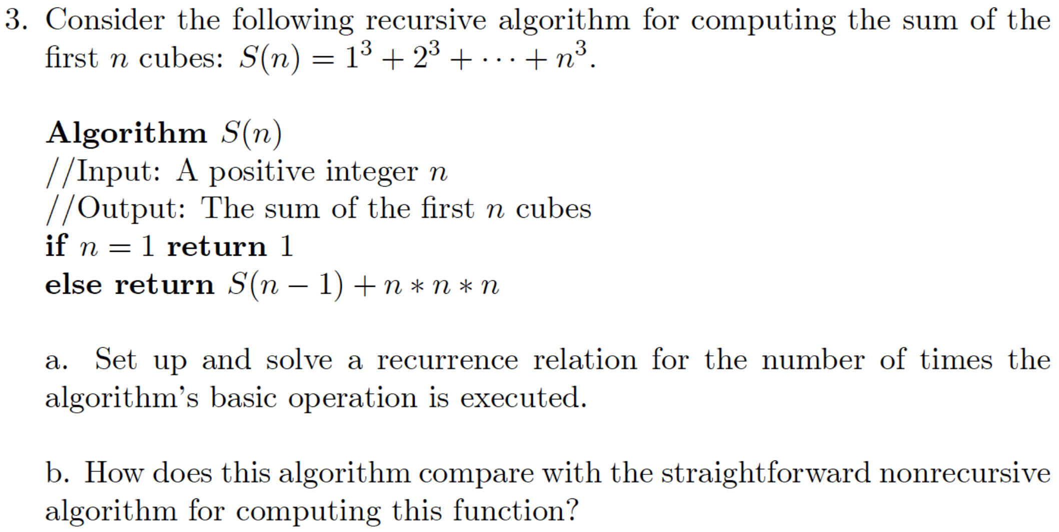
a. What does this algorithm compute?

b. What is its basic operation?

c. How many times is the basic operation executed?

d. What is the efficiency class of this algorithm?

Exercises 2.4 #3 (5 points)



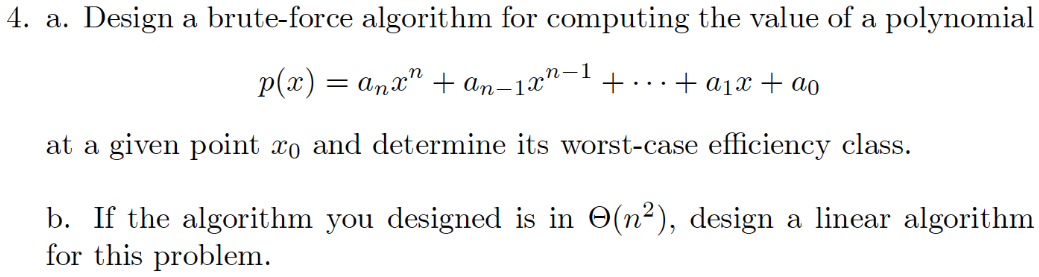
3. Consider the following recursive algorithm for computing the sum of the first n cubes: $S(n) = 1^3 + 2^3 + \cdots + n^3$.

```
**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.

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

Exercises 3.1 #4-a,b (5 points)



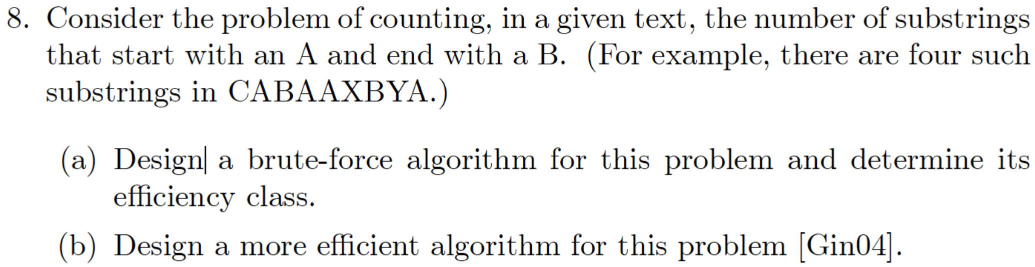
4. a. Design a brute-force algorithm for computing the value of a polynomial

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$$

at a given point x_0 and determine its worst-case efficiency class.

b. If the algorithm you designed is in $\Theta(n^2)$, design a linear algorithm for this problem.

Exercises 3.2 #8 (4 points)

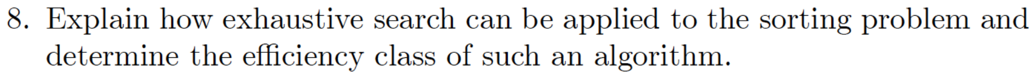


8. Consider the problem of counting, in a given text, the number of substrings that start with an A and end with a B. (For example, there are four such substrings in CABAAXBYA.)

   (a) Design a brute-force algorithm for this problem and determine its efficiency class.

   (b) Design a more efficient algorithm for this problem [Gin04].

Exercises 3.4 #8 (2 point)



8. Explain how exhaustive search can be applied to the sorting problem and determine the efficiency class of such an algorithm.

Note(s):

- Use C++ or Java for those problems that require algorithm design.
- All problems may be found in the Levitin textbook.

Submission:

- Deadline: Monday, 4/10/2023, 11:59pm
- Submit your solutions as a PDF on Canvas under HW #2