Assignment objectives

Practice determining *basic operation* and calculating the *running time* of (non-recursive) algorithms.

Notes about the math

Note 1: This assignment involves writing some math expressions. Please try to concentrate your mental energy on doing the actual math (and algorithms!), while minimizing the amount of time/effort you spend on the mechanics of actually producing this document. Microsoft Word skills are not a COMP 3760 course objective. With that in mind, you may provide your math here by any means that is stress-free for you:

- Do it right in here with the Word/Office equation editor
- Use any other math tool; paste/insert into this document
- Write your math on paper; scan or photograph; paste into this document
- Hand-write *all* of your answers and build a PDF out of your handwritten pages
- Other?

I ask only that your answers—of course—be legible!

Note 2: You will be asked to find the *closed-form expressions* for some summations, aka to *simplify* or *solve* the summations (reduce them to a function of n without sigma/sum formulas). Sometimes this will require some algebra. You are *not required* to show all of your work, but it is beneficial for you to do so. Sometimes, if you make mistakes, you can still receive some credit if you "had it" partway along. But if you only show the final answer, and it is wrong, partial credit is not possible.

Submission information

Due date: As noted on Learning Hub.

Submit the following to the drop box:

1. This worksheet, completed

There are no file-naming requirements.

Please do not zip or compress your submission.

Marking information

This assignment contains 5 problems worth a total of 20 points.

Problem 1 (3 points)

1a. What line contains the basic operation in the following pseudocode?

```
1.  Algorithm CF (n)
2.  p = 1
3.  for i = 1 to n do
4.      p = p * i
5.  endfor
6.  END
```

Answer 1a:

line 4 contains the basic operation p=p*i

1b. Write a summation formula that represents the number of times the basic operation will be performed.

Answer 1b:

$$\sum_{i=1}^{n} 1$$

1c. How many times is the basic operation performed (a function of n)?

Answer 1c:

n

Problem 2 (3 points)

2a. What line contains the basic operation in the following pseudocode?

```
1.   Algorithm CS (A[0..n-1])
2.   s = 0
3.   for i = 0 to n-1 do
4.       s = s + A[i]
5.   endfor
6.   return s
7.   END
```

Answer 2a:

line 4 contains the basic operation s = s + A[i]

2b. Write a summation formula that represents the number of times the basic operation will be performed.

Answer 2b:

$$\sum_{i=0}^{n-1} 1$$

2c. How many times is the basic operation performed (a function of n)?

Answer 2c: n

Problem 3 (3 points)

3a. What line contains the basic operation in the following pseudocode?

```
1.   Algorithm MTT (n)
2.   A = new array[0..n-1][0..n-1]
3.   for i = 0 to n-1 do
4.       for j = 0 to n-1 do
5.           A[i,j] = i * j
6.       endfor
7.   endfor
8.   return A
9.   END
```

Answer 3a:

Line 5 contains the basic operation A[i,j] = i*j

3b. Write a summation formula that represents the number of times the basic operation will be performed.

Answer 3b:

$$\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} 1$$

3c. How many times is the basic operation performed (a function of n)?

Answer 3c:

n^2

Problem 4 (4 points)

4a. What line contains the basic operation in the following pseudocode?

```
 1.   Algorithm IMS (A[0..n-1][0..n-1])
 2.   for i = 0 to n-1 do
 3.       if A[i,i] == 0 then
 4.           return false
 5.       endif
 6.   endfor
 7.   for r = 1 to n-1 do
 8.       for c = 0 to r-1 do
 9.           if A[r,c] != A[c,r] then
10.               return false
11.           endif
12.       endfor
13.   endfor
14.   return true
15.   END
```

Answer 4a:

line 9 contains the basic operation if A[r,c] != A[c,r]

4b. Write a summation formula that represents the number of times the basic operation will be performed.

Answer 4b:

$$\sum_{r=1}^{n-1} \sum_{c=0}^{n-1} 1$$

4c. How many times is the basic operation performed (a function of n)?

Answer 4c:

$$\frac{(n-1)n}{2}$$

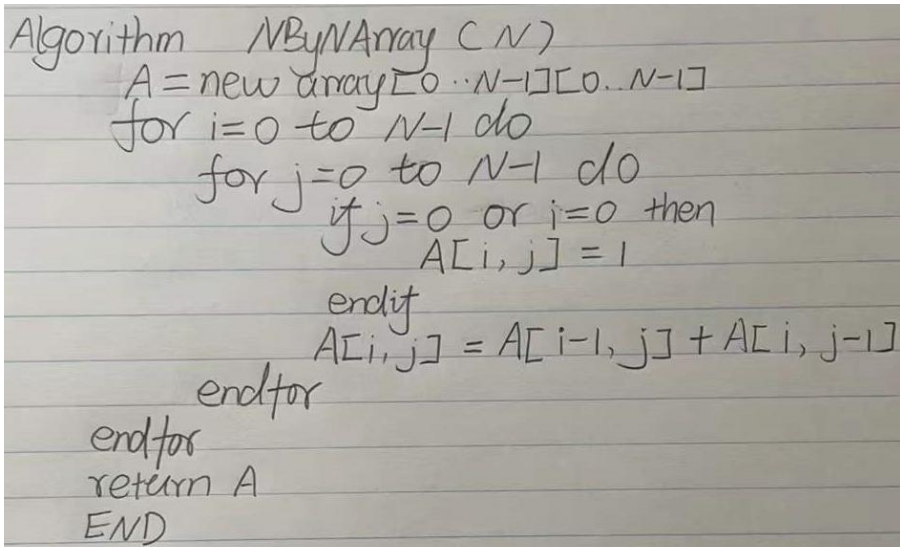
Problem 5 (7 points)

Write a pseudocode algorithm that solves the following problem:

Given an integer N, create and return a new NxN array of integers and initialize the array as follows. Every element in the first row (row index 0) is 1. Every element in the first column is 1. For all other elements in the array, the value is the sum of the element immediately above, and the element immediately to the left. For example, here is the result array for N=5:

	0	1	2	3	4
0	1	1	1	1	1
1	1	2	3	4	5
2	1	3	6	10	15
3	1	4	10	20	35
4	1	5	15	35	70

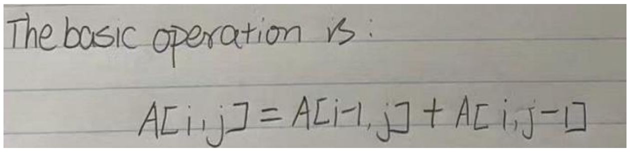
Answer 5a (add more space as needed):



```
Algorithm NByNArray ( N )
    A = new array[0 ..N-1][0..N-1]
    for i=0 to N-1 do
        for j=0 to N-1 do
            if j=0 or i=0 then
                A[i, j] = 1
            endif
            A[i, j] = A[i-1, j] + A[i, j-1]
        endfor
    endfor
    return A
END
```

5b. What is the basic operation of your algorithm?

Answer 5b:

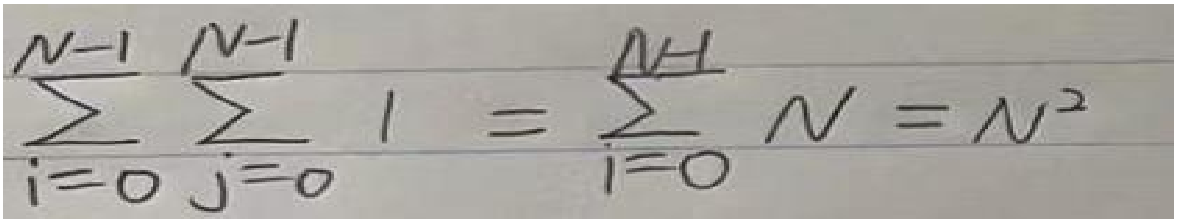


The basic operation is :

A[i, j] = A[i-1, j] + A[i, j-1]

5c. How many times is the basic operation performed (a function of n)?

Answer 5c:



$$\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} 1 = \sum_{i=0}^{N-1} N = N^2$$