What is allowed use during the exam:

- Textbook and notes (your own and/or the instructor's)
- Your own memory sheet(s)
- Your own code from the labs
- Windows calculator or pocket calculator

What is not allowed to use:

- Any web browser (except for using a web-based IDE for programming)
- Any input from other people

Please show all the steps in your solution. Credit is awarded for complete solutions, not just for the final answers. The final answers by themselves (especially when wrong!) will bring little credit.

I, the undersigned, understand the instructions above, and I will abide by them. I will not try to seek, use or provide any dishonest help.

Name (Print!): Edwin Sparks Signature: Edwin (Tony) Sparks

Pencil and paper problems

1] (20 pts.) An algorithm has an efficiency $C(N) = 3^N + 2$, where N is the size of the problem.

(a) (10 pts.) Assume that we have a computer that solves a problem of size **N** = 15 in 20 seconds using this algorithm. Calculate the time c_{op} in which the computer performs one fundamental operation c_{op}:

$C(n) = 3^n + 2 \approx 3^n$ $\quad C_{op} = 20$

$T(n) \approx C_{op} C(n) \quad T(n) \approx 20 \cdot$

$C(n) = 3^n$ $\quad$ ~~$C_{op} = 1.33 \text{ sec} = 1 \text{ op}$~~ $\quad \boxed{1.33 \text{ sec for } 1 \text{ op}}$?

~~$\frac{T(15n)}{T(n)} = \frac{C_{op}(C(15n))}{C_{op}(C(n))} = \frac{3^{15n}+2}{3^{n}+2} = \frac{3^{15n}}{3^{n}} = 15$~~

$\frac{T(15(n))}{T(n)} = \frac{3^{15n}}{3^n} = 15\times$ per op

(b) (9 pts.) Your boss it telling you to use the same computer and the same algorithm to solve a problem twice the size, meaning **N** = 30. How long will it take?

~~$\frac{15 \text{ ops}}{20 \text{ sec}} = \frac{30}{x} \quad \frac{15x}{15} = \frac{600}{15} \quad x = 40 \text{ sec}$~~

doubles would take 30x longer than 20sec

So $20 \times 30 = \frac{600}{60} = \boxed{10 \text{min}}$

(c) (1 pt.) What do you tell your boss?

10 mins for an input size N=30, Sir?

2] (15 pts.) The efficiency of an algorithm is given by this sum: C(N) = 15 + 20 + 25 + 30 + ... + 5·N, where N is the size of the problem. (N must be ≥ 3, do you see why?)

a. Using the appropriate summation formula from Appendix A, find a closed-form exact expression (no approximations!) for C(N):

$$\sum_{i=15}^{5n} i = \frac{5n(5n+\cancel{1}\,5)}{2} = \text{Some form of } \tfrac{1}{2}n^2$$

$$\cancel{\Rightarrow \frac{25n^2+5n}{2} = \tfrac{1}{2}25n^2+5n}$$

$$\Rightarrow \frac{5n(5n+5)}{2} = C(n)$$

b. Use the closed-form expression found above to approximate the function C(N) by "throwing away" all the low-order terms (but do keep the constant coefficient of the highest-order term!)

$$\frac{5n(5n+5)}{2} = \tfrac{1}{2}25n^2+25n \approx \tfrac{1}{2}25n^2 \approx C(n)$$

c. Use the approximation found above to calculate the order of growth Θ. Use the simplest function of N inside theta!

$\Theta(n^2)$

3] (10 pts.) In each case below, decide if the recursion is linear and homogenous (circle). If not, briefly explain why!

3rd order

a. M(n) = M(n-1) + M(n-2)·M(~~N~~ n-3) Linear? Yes / No Homogenous? Yes / No

Explain: m^2 $\neq C$ So Homogenous

b. A(n) = A(n-1) + A(n-2) + 2·A(n-3) Linear? Yes / No Homogenous? Yes / No

Explain: 2·A ✓ $\neq C$ ✓

c. X(n) = 42·X(n-1) - X(n-2) - X(n-4) Linear? Yes / No Homogenous? Yes / No

Explain: 42·X ✓ $\neq C$ ✓

d. B(n) = 3·B(n-1) + 5·B(n-2) + 2·n Linear? Yes / No Homogenous? Yes / No

Explain: ~~2·n would be a Constant?~~ n is from the Series.

~~So no Homogenous~~

e. C(n) = - C(n-2) + 10 Linear? Yes / No Homogenous? Yes / No

Explain:

Constant.

4] (15 pts.) This is the pseudocode of a non-recursive algorithm:

```
ALGORITHM fun( A[0..n-1][0..n-1] )
//Input: A two-dimensional array (matrix) of integers
//Output: A boolean value (true/false or 1/0)
for i ← 0 to n-1
   for j ← i+2 to n-1
      if A[i][j] ≠ 0
         return false
return true      i ++
                 j ++
```

(a) (1 pt.) What is the fundamental operation that describes its efficiency?

Answer: True, False, ~~So~~ Comparison ?

(b) (3 pts.) What is the best case for the algorithm efficiency?

Answer for the best case: $C_{best}(n)$ = 1

Explain when it happens: When $A[i][j] = 0$, it returns false

(c) (3 pts.) What is the worst case? Explain in your own words when it happens:

Worst case we go through every [j] element with every [i] element. And never return True

(d) (4 pts.) Set up a summation for the worst case, using the summation symbol:

(1 op)

$C_{worst}(n) = \sum_{i=0}^{n-1} \sum_{j=i+2}^{n-1} 1$

(e) (4 pts.) Solve the summation to find an exact expression (no approximations!) for $C_{worst}(n)$:

$$-\sum_{i=0}^{n-1} \sum_{j=i+2}^{n-1} 1 = \sum_{i=0}^{n-1} [n-1-i-2+1] = \sum_{i=0}^{n-1} (n-i-2) = (n-i-2) \sum_{i=0}^{n-1} 1$$

$$\Rightarrow (n-i-2) \cdot [n-1-0+1] = (n-i-2)(n) = \underline{n^2 - in - 2n},$$

order of growth $\Theta(n^2)$

Final answer for the exact expression: $C_{worst}(n) = n^2 - in - 2n$

(Mostly) Programming problems

5] (20 pts. total) Two algorithms have the following count functions: $C_1(n) = 42 \cdot n \cdot \sqrt{n}$ and $C_2(n) = 2 \cdot n^2$.

(a) Obviously, the second algorithm has the largest order of growth, but someone says: 'For n=1, C_1 gives 42 basic operations, whereas C_2 gives only 2, therefore the first algorithm must have a larger nr. of basic operations.'

How do you reply?

Answer (2 pts.): You are correct the first algorithm does have a higher number of basic operations in lower numbers, which is what C(n) Calculates and has nothing to do with order of growth. Ignore Constants and Simplify!

$C_1 = n^{4/3}$ $C_2 = n^2$ $C_1 < C_2$

$\sqrt[3]{4n}$

(b) (17 pts.) Write a C program to verify the behavior you described above, and to find the intersection point for the functions. The program should use a loop to display on each line the values of n, $C_1(n)$, and $C_2(n)$. (You may add the ratio if you are following the code from the lab.) Use an increment of 1 in the loop.

►Attach screenshots of your source code and output

(c) Use the output from the program to answer the question: What is the first integer n for which $C_2(n)$ is larger than $C_1(n)$?

Answer (1 pt.): 441

$$\frac{42 \cdot n \cdot \sqrt{n}}{n} = \frac{2n^2}{n}$$

$$\frac{42\sqrt{n}}{\sqrt{n}} = \frac{2n}{\sqrt{n}}$$

$$42 = \frac{2n}{n^{1/2}}$$

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help Midterm_1 - main.c

Midterm_1 〉 main.c

Project

- Midterm_1 E:\Tarleton Fall 21'\Data Structures Lab\Midte
- External Libraries
- Scratches and Consoles

main.c

```c
// Sonny Sparks
// Data Structures
// Midterm 1
// September 22, 2012

#include <stdio.h>
#include <math.h>
/////////////////////////////////////////////////
int main()
{
    double n=10000000, C1, C2;
    for(int i=0; i<=n;i++) {
        C1 =42*i*sqrt(i) ;
        C2 =2*(i*i);
    if(C1 == C2)
        printf( _Format: "The intersection of C1 and C2 is: %d", i);
    }
    return 0;
}
```

main

Run: Midterm_1

```
"E:\Tarleton Fall 21'\Data Structures Lab\Midterm_1\cmake-build-debug\Midterm_1.exe"
The intersection of C1 and C2 is: 0The intersection of C1 and C2 is: 441
Process finished with exit code 0
```

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help Midterm_1 - main.c

Midterm_1 〉 main.c

Project

- Midterm_1 E:\Tarleton Fall 21'\Data Structures Lab\Midte
- External Libraries
- Scratches and Consoles

main.c

```c
// Sonny Sparks
// Data Structures
// Midterm 1
// September 22, 2012

#include <stdio.h>
#include <stdbool.h>
#define N 5
///////////////////////////////////////////////////
int A[N][N]={{1, 10, 0, 20, 0},{2, 2, 20, 0, 0},{3, 4, 3, 0, 0},{4, 5, 10, 4, 0},{5, 6, 20, 30, 5}};
///////////////////////////////////////////////////
bool fun(){
    for(int i=0;i<N-1;i++){
        for(int j=i+2;j<N-1;j++)
            if(A[i][j]!=0)
                printf("i:%d\tj:%d\tFalse\n", i, j);
            else
                printf("i:%d\tj:%d\tTrue\n", i, j);
    }
}
///////////////////////////////////////////////////
int main()
{
    fun();
    return 0;
}
```

fun

Run: Midterm_1

```
"E:\Tarleton Fall 21'\Data Structures Lab\Midterm_1\cmake-build-debug\Midterm_1.exe"
i:0     j:2     True
i:0     j:3     False
i:1     j:3     True
```