Adam Jablonski

CS-260 HW1

The following pages are my attempt at problem 1.12. I did look at 1.10 however I was not able to grasp the concept well enough to complete the homework assignments. I also didn’t have a solid understanding of 1.12 however I did attempt it based on intuition and the amount I did understand.

Adam Jablonski

CS-260

1.12

a) procedure matmpy (n: integer);

```
var
    i, j, k: integer
begin
    for i:=1 to n do
        for j:=1 to n do begin
            C[i, j]:=0;
            for k:=1 to n do
                C[i, j]: C[i, j] + A[i, k] * B[k, j]
        end
    end
```

$O(n^3)$ Since there are 3 for loops iterating up till n

b) procedure mystery (n: integer)

```
var
    i, j, k: integer
begin
n:                                    for i:=1 to n-1 do    }* are equal because i goes to n-1
$\sum_{j=2}^{n} j$:                    for j:=i+1 to n do    }  and j:=i+1 goes to n
$\sum_{k=3}^{n+1}\sum_{j=k}^{n+1} j$:      for k:=1 to j do
                                              { .... O(1) }
    end
```

$$n + \sum_{j=2}^{n} j + \sum_{k=3}^{n+1}\sum_{j=k}^{n+1} j = O(n^3)$$

c) $n + n(2n+5)(n-1)$ +

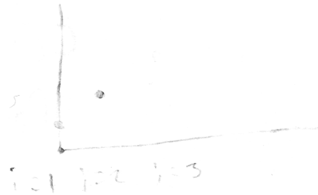
C) Procedure very odd (n: integer);

```
var
  i, j, x, y: integer

begin
  for i: =1 to n do
    if odd(i) then begin
      for j: i to n do
        x: = x+1;

        for j: =1 to i do
        y: = y+1
      end
    end
```

Assume its odd

$O(1) = n$

$$\sum_{j=1}^{n} \sum_{j=1}^{i} = n^2$$



$\boxed{O(n^3)}$

D) It is recursive.

```
function recursive (n: integer) : integer;
  begin
    if n<=1 then
      return(1)
    else
      return (resursive(n-1) + recursive (n-1))
  end
```

$T(0) = O(1)$

$= \{\text{recursive}(n-1) + \text{recursive}(n-1)\} = 2\{\ldots\}$

$\boxed{= O(2^n)}$