Joey Amand

CSC 6013 - Worksheet for Week 4

1) Complete this table rounding each decimal to the nearest integer. This should give you a sense of the comparative growth rates of the functions. Note that *lg(n)* means python's log2(n).

$\lg(n)$	$\sqrt{n}$	n	$n\lg(n)$	n^2	n^3	2^n	$n!$
0	1	1	0	1	1	2	1
1	1	2	2	4	8	4	2
2	2	3	5	9	27	8	6
2	2	4	8	16	64	16	24
2	2	5	12	25	125	32	120
3	2	6	16	36	216	64	720
3	3	7	20	49	343	128	5040
3	3	8	24	64	512	256	40320
3	3	9	29	81	729	512	362880
3	3	10	33	100	1000	1024	3628800

For #2, 3, and 4, solve each summation in two ways: write out all terms of the sum and perform the arithmetic; then use one of the formulas in the class notes to confirm your answer.

2) $(1)^2+(2)^2+(3)^2+(4)^2+(5)^2+(6)^2+(7)^2+(8)^2+(9)^2+(10)^2+(11)^2+(12)^2+(13)^2+(14)^2+(15)^2$

$\sum_{i=1}^{15} i^2$ $1+4+9+16+25+36+49+64+81+100+121+144+169+196+225$

$= 1240$

$= \frac{n(n+1)(2n+1)}{6} = \frac{15(16)(31)}{6} = 1240$

3) $2^0+2^1+2^2+2^3+2^4+2^5+2^6+2^7+2^8+2^9+2^{10}$

$\sum_{i=0}^{10} 2^i$ $1+2+4+8+16+32+64+128+256+512+1024 = 2047$

$= 2^{n+1}-1 = 2^{11}-1 = 2047$

4) Follow the same instructions for #2 and #3; round off this answer to three decimal places.

$\sum_{i=1}^{8} 2^{-i}$ $2^{-1}+2^{-2}+2^{-3}+2^{-4}+2^{-5}+2^{-6}+2^{-7}+2^{-8}$

$\frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{16}+\frac{1}{32}+\frac{1}{64}+\frac{1}{128}+\frac{1}{256} = \frac{255}{256} = 0.996$

$= 1-2^{-n} = 1-\frac{1}{2^n} = 1-\frac{1}{256} = \frac{255}{256} = 0.996$

For #5, solve this summation in two ways: write out all terms of the sum, but there will be no arithmetic to perform; then use one of the formulas in the class notes to express the sum as a polynomial function of n.

5) $= 1+2+3+\ldots+(n-1)$

$$\sum_{i=1}^{n-1} i = \frac{(n-1)(n)}{2} = \frac{n^2-n}{2} = \frac{1}{2}n^2 - \frac{1}{2}n$$

Asymptotic analysis for #6 and #7: Determine the Big-Oh class of each algorithm. That is, **formally** compute the worst-case running time as we did on class using a table to produce a function that tracks the work required by all lines of code. Include all steps of the algebraic simplification, but you do not need to provide comments to justify each step.

6) Arithmetic mean = "add them all up, and divide by how many". Let the size of the problem = n = the number of entries in the array.

```
1   # Input: an array A of real numbers
2   # Output: the arithmetic mean of the entries in the array
3   def arithmeticMean(A):
4       sum = 0  C1 · 1
5       count = 0 C2 · 1
6       for x in A: C3 · n
7           sum += x  C4 · n
8           count += 1 C5 · n
9       average = sum/count C6 · 1
10      return average  C7 · 1
```

line 4 c_1 1
line 5 c_2 1
line 6 c_5 n
line 7 c_4 n
line 8 c_5 n
line 9 c_6 1
line 10 c_7 1

$T(n) = c_1 + c_2 + c_3 n + c_4 n + c_5 n + c_6 + c_7$

$T(n) = (c_3 + c_4 + c_5)n + (c_1 + c_2 + c_6 + c_7)$

$T(n) = c_8 n + c_9$

$T(n) \leq c_8 n + c_9 n$ when $n \geq 1$

$T(n) \leq c_{10} n$

$T(n)$ is in Big-Oh class $= O(n)$

7) Sum of entries in an upper triangular nxn array. Let the size of the problem = n = the dimension of the nxn matrix.

```
1   # Input: an upper triangular square matrices A where all entries below the diagonal = 0,
2   #        and an integer n giving the dimension of this nxn matrix
3   # Output: a real number giving the sum of the entries
4   def UpperTriangularMatrixSum (A, n):
5       sum = 0  C1 · 1
6       for i in range(0, n): C2 · n
7           for j in range(i, n): C3 · Σ_{i=0}^{n} i = C3 · n(n+1)/2
8               sum += A[i][j] C4 · n(n+1)/2
9       return sum C5 · 1
```

line 5 c_1 1
line 6 c_2 n
line 7 c_3 $\frac{n(n+1)}{2}$
line 8 c_4 $\frac{n(n+1)}{2}$
line 9 c_5 1

$T(n) = c_1 + c_2 n + c_3\left(\frac{n(n+1)}{2}\right) + c_4\left(\frac{n(n+1)}{2}\right) + c_5$

$c_3\left(\frac{1}{2}n^2 + \frac{1}{2}n\right) + c_4\left(\frac{1}{2}n^2 + \frac{1}{2}n\right)$

$T(n) = c_1 + c_2 n + \frac{c_3}{2}n^2 + \frac{c_3}{2}n + \frac{c_4}{2}n^2 + \frac{c_4}{2}n + c_5$

$T(n) = (c_1 + c_5) + \left(c_2 + \frac{c_3}{2} + \frac{c_4}{2}\right)n + \left(\frac{c_3}{2} + \frac{c_4}{2}\right)n^2$

$T(n) = c_6 + c_7 n + c_8 n^2$

$T(n) \leq c_6 n^2 + c_7 n^2 + c_8 n^2$ when $n \geq 1$

$T(n) \leq c_9 n^2$

$T(n)$ is in Big-Oh class $= O(n^2)$