Lab wk1-2: Asymptotic notation and finding closed form solutions

Deliverable: Submit you answer sheet at the end of lab

Asymptotic notation:

1. a. Consider the standard algorithm for adding two n x n matrices. What is its basic operation? How many times is it performed as a function of the matrix order – n ? As a function of the total number of elements in the input matrices?
   b. Answer the same questions for the standard algorithm for matrix multiplication.

2. For each of the following functions, indicate how much the function's value will change if its argument is increased by a factor of 4.
   a. $\log_2 n$
   b. sqrt(n)
   c. n
   d. n^2
   e. 2^n

3. What is the **best** way to describe relationship between each pair of functions
   e.g. f(n) is O(g(n)), Ω(g(n)), or Θ(g(n))?

f(n)	g(n)	f(n) is O(g(n)), Ω(g(n)), or Θ(g(n))
$\log_2 n$	$n^{1/2}$	
n	$n^{1/2}$	
n^k where k>1	n!	
$(\log_2 n^n)$	$n^{3/2}$	
n/2	n	
n^k	n!	
n^2	$(\log_2 2^n)*(n/2)$	
$\log_2(n^{n+1})$	n^2	
n^k where k>1	c^n where c>1	
2^n	2^{n+1}	

Two Important Summation Formulas

Arithmetic series

$\sum_{i=1}^{n} i = \frac{n*(n+1)}{2}$ This generalizes to $\sum_{i=1}^{n} Ci$ =C $\frac{n*(n+1)}{2}$

Geometric series

$\sum_{i=0}^{n} a^i = \frac{a^{n+1}-1}{a-1}$ a≠ 1 when a =2 then $\sum_{i=0}^{n} 2^i = \frac{2^{n+1}-1}{2-1} = 2^{n+1} - 1$

4. Compute the following sums.
   a. 1 + 3 + 5 + 7 + · · · + 999
   b. 2 + 4 + 8 + 16+· · · + 1024

5. Mental arithmetic A 10×10 table is filled with repeating numbers on itsdiagonals as shown below. Calculate the total sum of the table's numbers in your head.

1	2	3			•••			9	10
2	3						9	10	11
3						9	10	11	
					9	10	11		
				9	10	11			
⋮			9	10	11				⋮
		9	10	11					
	9	10	11						17
9	10	11						17	18
10	11				•••		17	18	19

Solving Recurrence Relations

6. Solve the following recurrence relations.
   a) $t(n) = t(n - 1) + 5$ for $n > 1$; $t(1) = 0$

   b) $t(n) = 3\, t(n - 1)$ for $n > 1$; $t(1) = 4$

   c) $t(n) = t(n - 1) + n$ for $n > 0$; $t(0) = 0$

   d) $t(n) = t(n / 2) + n$ for $n > 1$; $t(1) = 1$ (solve for $n = 2^k$)

7. Set up and solve a recurrence relation for the number of calls made by F(n), the recursive algorithm for computing n!