HOMEWORK 1

100 points

DUE DATE: January 26th 9:00pm.

Homework 1 Deliverables:

Please use a text editor and type your answers using only one text file, convert that text file to PDF upon submission. Do not submit multiple files (Example: Hw1.pdf should contain solutions to all 4 problems, we do not want to see Problem1.pdf, Problem2.pdf…. being uploaded on D2L).
Name your file as:
hw1_yournetid.pdf (replace yournetid with the actual netid)

Warning: There should be only one file uploaded for this homework assignment. All other files will not be considered for grading. This is to prevent students from uploading multiple files.
Thank you.
This is not a team work, do not copy somebody else's work.

Your hw1_yournetid.PDF must be submitted via D2L Dropbox no later than January 26th 2017 by 9:00pm

Problem 1

Show that

$$n^{\log_b m} = m^{\log_b n}$$

for any b >0, n>0, m>0.

Problem 2

a) When writing binary numbers, you may have noticed the pattern of numbers that are written with all-ones:

$$3\,(=4-1=2^2-1)=(11)_2$$

$$7(=8-1=2^3\;-1)=(111)_2$$

$$15=(1111)_2$$

and so on. The pattern being that $2^n - 1$ is written in binary as *n* ones; from the formal definition of binary notation, *use the formula for the geometric sum* to verify the above pattern (i.e., to show that the pattern holds for any arbitrary n).

b) In decimal notation, *n* nines represent the value $10^n - 1$; using the same geometric sum formula, show that the pattern holds for any arbitrary *n*.

Problem 3:

a) Using the following figure shown below as your guide (Reference: Introduction to Algorithms, Ch2, T. Cormen, C. Leiserson, R. Rivest, C. Stein), **illustrate** the operation of INSERTION-SORT on the array A = [31, 41,59, 26, 41, 58].

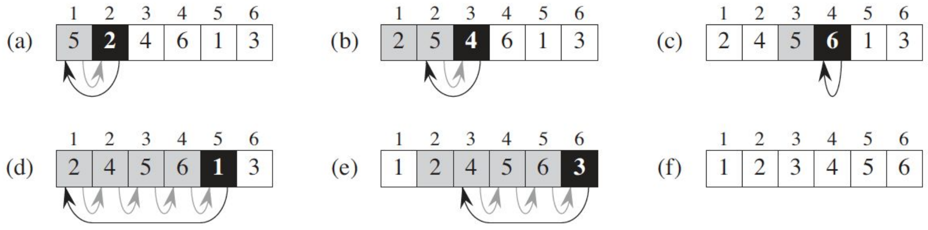
(a) 1 2 3 4 5 6 | 5 2 4 6 1 3
(b) 1 2 3 4 5 6 | 2 5 4 6 1 3
(c) 1 2 3 4 5 6 | 2 4 5 6 1 3
(d) 1 2 3 4 5 6 | 2 4 5 6 1 3
(e) 1 2 3 4 5 6 | 1 2 4 5 6 3
(f) 1 2 3 4 5 6 | 1 2 3 4 5 6

Figure 2.2 The operation of INSERTION-SORT on the array $A = \langle 5, 2, 4, 6, 1, 3 \rangle$. Array indices appear above the rectangles, and values stored in the array positions appear within the rectangles. **(a)–(e)** The iterations of the **for** loop of lines 1–8. In each iteration, the black rectangle holds the key taken from $A[j]$, which is compared with the values in shaded rectangles to its left in the test of line 5. Shaded arrows show array values moved one position to the right in line 6, and black arrows indicate where the key moves to in line 8. **(f)** The final sorted array.

b) Rewrite the INSERTION-SORT procedure discussed in class to sort data into decreasing order. Use the same pseudocode style discussed and demonstrated in class. (See Lecture 2 January 12th folder on D2L for additional guidance)

Problem 4:

Assuming that you have already completed the linear search method for Hw 0, you will now analyze your algorithm. In short terms Linear search (using an array or a vector) is:

Input: A sequence of *n* numbers
and a value *v*.
Output: An index *i* such that:

or the special value *NIL if v* does not appear in A.

a)Write a pseudocode for your Linear Search (Using the same style discussed in class, see previous problem for more information)

Name your array as **A** and you can use the length property of your array for example

```
h ← length[A]
```

Place a line number on each line of code for example

```
1. h ← 0
2. while h<= length[A] …
3. …..
```

b)What is the run time analysis of your algorithm? Complete the cost-time analysis to solve T(n). When analyzing your code place a cost and times column next to your pseudo code as demonstrated in class.When solving for T(n) do not just type the end result, show us your steps that lead you to the result. No credit will be given if you just enter the result without showing steps. (See Insertion sort cost time analysis covered in class if you need more guidance)

For example:

	Cost	Times
1. h ← 0	c_1	1
2. while h<= length[A] …	c_2	….
3. ….		

$T(n)= c_1 *1 + c_{2*}(....)......$

1. How many elements of the input sequence need to be checked on the average, assuming that the element being searched for is equally likely to be any element in the array? Explain.
2. What is the best case for linear search?Explain.
3. What is the worst case? Explain.
4. What are the average-case and worst case running times of linear search in Θ notation. Justify your answer.