ADA Assignment-1

Q1. Describe the method for analyzing an algorithm. What do you mean by Best case, Average case, Worst case time complexity of an algorithm?

Q2. What is growth of function? Explain the following term:

(a) Big oh Notation(O)
(b) Big Omega Notation(Ω)
(c) Big Theta Notation(Θ)
(d) Little Oh Notation(o)
(e) Little Omega Notation(ω)

Q3. Find the Big theta Notation for the following functions:

(a) f(n)= $79n^3+44n$
(b) f(n)=$2 \times 2^n + 6n^2 + 3n$
(c) f(n)=$27n^2 + 16n + 25$

Q4. Find the Big oh(O) Notation for the following functions:

(a) f(n) = $10n^2 + 7$
(b) f(n) = $5n^3 + n^2 + 3n + 2$
(c) f(n) = $3 \times 2^n + 4n^2 + 5n + 3$

Q5. Find the Big oh(Ω) Notation for the following functions:

(a) f(n) = $27n^2 + 16n$
(b) f(n) = $3n^3 + 4n$
(c) f(n) = $2^n + 6n^2 + 3n$

Q6. Prove the following:

(a)) $\lg n! = \theta(n \lg n)$
(b) $n! = 0(n^n)$
(c) $\sum_{i=1}^{n} \log(i) = \theta(n \lg n$

Q7. Consider the polynomial in n of the form

f(n) = $\sum_{i=0}^{m} a_i n^i = a_m n^m + a_{m-1} n^{m-1} + \cdots + a_2 n^2 + a_1 n + a_0 \; where \; a_m > 0.$

then f(n) = $\Omega(n^m)$.

Q8. For the given x and n ,write an algorithm to compute $x^n/_{n!}$,and compute the running time.

ADA Assignment 2

Solve the recurrence by Substitution method:

Q1. $T(n) = 2T\left(\frac{n}{2} + 17\right) + n$

Q2 $T(n) = T\left(\frac{n}{2}\right) + 1$

Q3. Solve the recurrence $T(n) = 2T\sqrt{n} + 1$

Solve the recurrence by Iteration method:

Q4. $T(n) = 3T\left(\frac{n}{2}\right) + n$

Q5. $T(n) = T(\alpha n) + T[(1-\alpha)n] + n$

Q6. $T(n) = T(n-1) + n$

Q7. $T(n) = T(n-1) + \frac{1}{n}$

Q8. $T(n) = T\left(\frac{n}{3}\right) + n^k$

Q9. $T(n) = 2T\left(\frac{n}{2}\right) + n^2$

Q10. $T(n) = \begin{cases} c, & if\ n = 1 \\ T(n-1) + T(n-1) + c, & ifn > 1 \end{cases}$

Solve the recurrence using Recursion tree:

Q11 $T(n) = 2T\left(\frac{n}{2}\right) + n^2$

Q12. $T(n) = T\left(\frac{n}{3}\right) + T\left(\frac{2n}{3}\right) + n$

Q13. $T(n) = T(n-a) + T(a) + cn \ \ where\ a \geq 1, c > 0$

Solve the recurrence using Masters Method:

Q14. $T(n) = 3T\left(\frac{n}{4}\right) + n \lg n$

Q15. $T(n) = 2T\left(\frac{n}{2}\right) + n^3$

Q16. $T(n) = 3T\left(\frac{n}{2}\right) + 2n^{1.5}, T(1) = 1.$

Q17. $T(n) = 2\,T\left(\frac{n}{4}\right) + \sqrt{n}$

Q18. $T(n) = 16T\left(\frac{n}{2}\right) + (n\lg n)^4$

Q19. $T(n) = 9T\left(\frac{n}{3}\right) + n^3 \lg n$

Q20. $T(n) = T\left(\frac{n}{2}\right) + \theta(1)$

Assignment 3

Q1: Construct the string matching automaton for the pattern P=aabab and illustrate its operation on text T=aaababaabaabab.

Q2: Taking modulo q=11, how many spurious hits does Rabin Karp matcher encounter in text T= 314151718923 when looking for pattern P=51.

Q3:What is metroid, prove that undirected graph is a metroid?

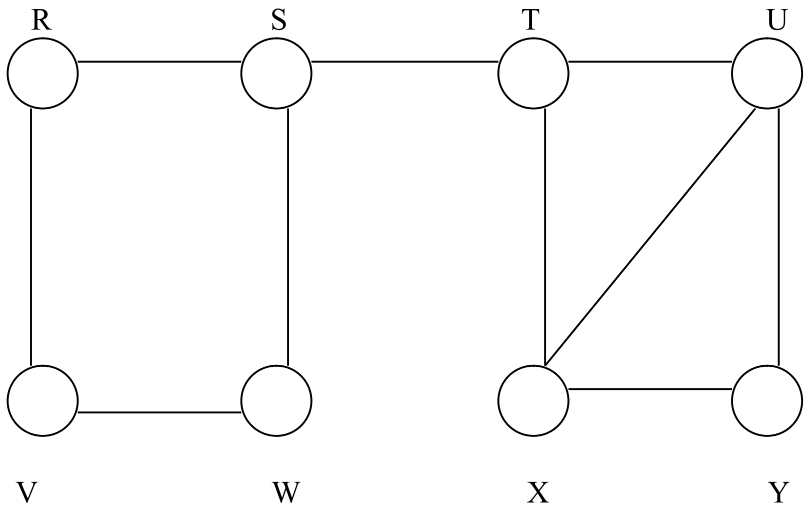
Q4: Write down the Parenthesis Theorem.

Q5:Using Knuth Morris Pratt algorithm find the prefix function for P=aab.

Q6: Prove that a directed graph G is acyclic iff a depth first search of G yields no back edges.

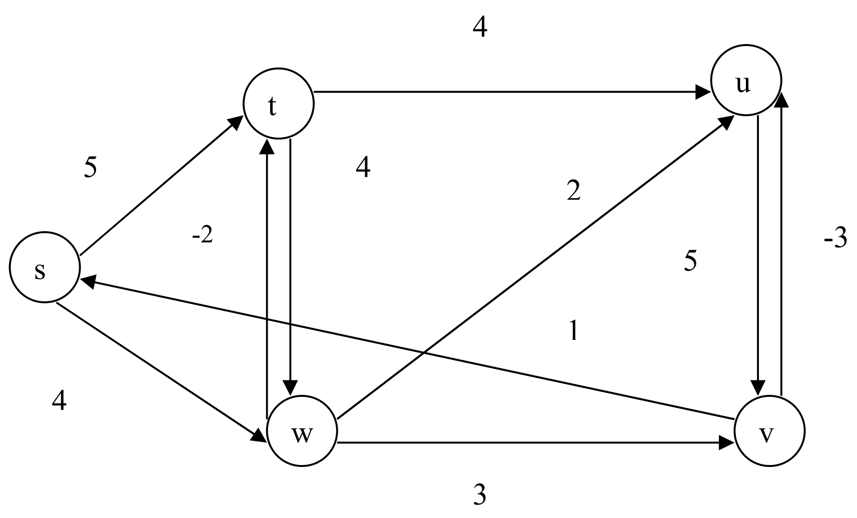
Q7: Prove that if two vertices are in the same strongly connected component then no path between them ever leaves strongly connected component.

Q8:



Find the breadth first search on the above graph using u as the source vertex.

Q9:



Find the shortest distances for all vertices of the above graph using s as the source.

Q10: Can Dijkstra's algorithm be applied on the above graph?

Q11: Apply Floyd Warshall algorithm on the above graph.

Q12: What is the difference between Prim and Kruskal's algorithm to find the minimum spanning tree?

Q13: Write the naïve string matching algorithm and discuss its complexity.

Q14: What is suffix function?

<u>ADA Assignment 4</u>

1)Give the Formal Definition of Polynomial, NP-Complete and NP-Hard Problems.

2) Difference between Optimization and Decision Problems.

3) Show that for any problem in NP, there is an algorithm which solves π in time O(2) p(n)), where n is the size of the input instance and p(n) is a polynomial (which may depend on π).

4) The Longest Path problem takes as input a simple graph G = (V, E) and a nonnegative integer k and decides if G has a simple path that is at least as long as k. Assuming that the Hamilton Cycle problem is NP-Complete, prove that Longest Path is also NP-Complete.

5). Let DOUBLE-SAT denote the language {hFi | Boolean formula F has two or more satisfying assignments}. Prove that DOUBLE-SAT is an NP-complete language.

6) Show that, if problem X is NP-complete, then the complement of X is Co-NP complete.

7)Define Circuit-SAT problem. Show that CIRCUIT-SAT satisfies definition of NP-complete.

8)Define Hamiltonian Circuit Problem. Show that this problem satisfies definition of NP-complete.

9)Define reducibility.

10)Give the NP-Completeness Proof.