Design and Analysis of Algorithms
Session: Jan-May, 2020, (Course ID: CS2800)
Assignment 1: Incremental Design

Date of Submission : 3 Feb 2020

1. (a) Given that there is a number present in all the three sorted arrays $x[1] \leq \cdots \leq x[p], y[1] \leq \cdots \leq y[q], z[1] \leq \cdots \leq z[r]$,find it (atleast one of them if there are many common elements) in $O(p+q+r)$ time.

(b) Suppose we do not know in advance that such a common element exists. Determine whether or not it exists and locate it(atleast one) when it does.

2. Generalisation : Given a matrix $A[1 \cdots n][1 \cdots m]$ where each row is sorted and there is an element common in all rows, find its position.

Naive algorithm generalising problem 1 will be $O(nm^2)$. Can you obtain a $O(nm)$ algorithm?

3. (a) Two arrays $A[1 \cdots p]$ and $B[1 \cdots q]$ are strictly increasing.Find the number of common elements in both. That is number of t such that $t = A[i] = B[j]$ for some i, j.

(b) How is the solution altered if $A[1] \leq A[2] \leq \cdots \leq A[p]$ and $B[1] \leq B[2] \leq \cdots \leq B[q]$, that is, the arrays are non-decreasing rather that strictly increasing.

4. If $A(x) = a_n x^n + \cdots + a_1 x + a_0$, then the derivative of $A(x), A'(x) = n a_n x^{n-1} + \cdots + a_1$. Devise an algorithm which produces the value of a polynomial and its derivative at a point $x = v$. Determine the number of required arithmetic operations.

5. Let $A(x) = a_n x^n + \cdots + a_0, p = n/2$ and $q = \lfloor n/2 \rfloor$. Then a variation of Horner's rule states that

$$A(x) = (\cdots(a_{2p}x^2 + a_{2p-2})x^2 + \cdots)x^2 + a_0 + ((\cdots(a_{2q-1}x^2 + a_{2q-3})x^2 + \cdots)x^2 + a_1)x$$

Show how to use this formula to evaluate $A(x)$ at $x = v$ and $x = -v$.

6. Given the polynomial $A(x)$ as above in problem 5, devise an algorithm which computes the coefficients of the polynomial $A(x + c)$ for some constant c.

7. Given n points $(x_i, y_i), 1 \leq i \leq n$, devise an algorithm which computes both the interpolating polynomial $A(x)$ and its derivative at the same time. How efficient is your algorithm?

8. Given n points (x_i, y_i), our task is to find the coefficients of the unique polynomial $A(x)$ of degree $\leq n - 1$ which goes through these n points. Mathematically the answer to this problem was given by Lagrange

$$A(x) = \textstyle\sum_{1 \leq i \leq n} \left(\prod_{i \neq j, 1 \leq j \leq n} \frac{(x - x_j)}{(x_i - x_j)\text{ffl}} \right) y_i$$

(a) Verify that $A(x)$ satisfies the n points.

(b) Analyse and prove that the naive algorithm for Lagrange interpolation takes $O(n^3)$ time.

(c) Let $G_{j-1}(x)$ interpolate $j - 1$ points $(x_k, y_k) 1 \leq k \leq j$ such that $G_{j-1}(x_k) = y_k$. Also let $D_{j-1}(x) = (x - x_1) \cdots (x - x_{j-1})$. Then we can compute $G_j(x)$ by the formula

$$G_j(x) = (y_j - G_{j-1}(x_j))(D_{j-1}(x)/D_{j-1}(x_j)) + G_{j-1}(x)$$

Using this formula, the algorithm for computing the interpolating polynomial takes $O(n^2)$ time . Verify and prove the same.

9. A group of n persons have an independent information for gossip known only to himself. Whenever a person calls another person in the group, they exchange all the gossip information they know at that time of calling. What is the minimum number of calls they have to make in order to ensure that everyone of them knows all the information.

10. Give a linear-time algorithm that takes as input a directed acyclic graph $G = (V, E)$ and two vertices s and t, and returns the number of simple paths from s to t in G. For example, the directed acyclic graph of Figure 1 contains exactly four simple paths from vertex p to vertex v: *pov, poryv, posryv* and *psryv.* (Your algorithm needs only to count the simple paths, not list them.)

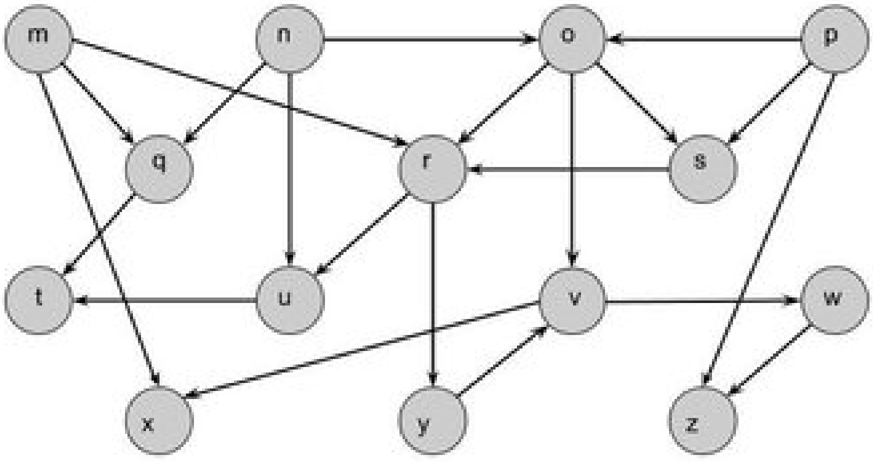

Figure 1: A DAG

11. Let G = (V,E) be a directed graph in which each vertex $u \in V$ is labeled with a unique integer L(u) from the set $\{1, 2, ..., |V|\}$. For each vertex $u \in V$, let R(u) = $\{v \in V : u \rightsquigarrow v\}$ be the set of vertices that are reachable from u. Define min(u) to be the vertex in R(u) whose label is minimum, i.e., min(u) is the vertex v such that L(v) = min$\{L(w) : w \in R(u)\}$. Give an O(V + E) time algorithm that computes min(u) for all vertices $u \in V$.

12. Let P be a simple, but not necessarily convex, polygon and q an arbitrary point not necessarily in P. Design an efficient algorithm to find a line segment originating from q that intersects the maximum number of edges of P. In other words, if standing at point q, in what direction should you aim a gun so the bullet will go through the largest number of walls. A bullet through a vertex of P gets credit for only one wall. An *O(n log n)* algorithm is possible.

13. Consider an $n \times n$ array A containing integer elements (positive, negative, and zero). Assume that the elements in each row of A are in strictly increasing order, and the elements of each column of A are in strictly decreasing order. (Hence there cannot be two zeroes in the same row or the same column.) Describe an efficient algorithm that counts the number of occurrences of the element 0 in A. Analyze its running time.

14. Design an incremental algorithm that constructs a permutation given it's inversion vector. What is the complexity of your algorithm? Can you design an *O(nlogn)* solution for the same?

15. (a) Let n=2^k-1, k$\geq$1. Find out exact number of comparisons done to build heap

    i. in a top-down fashion.

    ii. in a bottom-up fashion.

(b) Show that if heapsort is performed after building the heap in bottom up fashion, it's complexity is *2nlog(n+1)-2n* (for n=2^k-1, k$\geq$1). Construct an input that makes this many comparisons (building worst case examples).

16. The **transitive closure** G^t of a directed graph G is a directed graph with the same vertices as G, that contains any edge $u \to v$ if and only if there is a directed path from u to v in G. A **transitive reduction** of G is a graph with the smallest possible number of edges whose transitive closure is G^T.The same graph may have several transitive reductions.

(a) Describe an efficient algorithm to compute the transitive closure of a given directed graph.

(b) Prove that a directed graph G has a *unique* transitive reduction if and only if G is acyclic.

(c) Describe an efficient algorithm to compute a transitive reduction of a given directed graph.

17. Suppose we are given a directed acyclic graph G with a unique source s and a unique sink t. A vertex v $\notin$ s,t is called an **(s,t)-cut vertex** if every path from s to t passes through v, or equivalently, if deleting v makes t unreachable from s. Describe and analyze an algorithm to find every (s,t)-cut vertex in G.

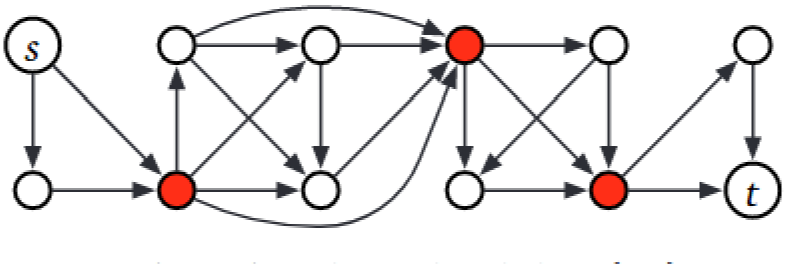

Figure 2: A directed acyclic graph with three (s,t)-cut vertices