Final Exam. (M.Tech) Data Structures and Algorithms.

1 (a) State Matrix chain multiplication Problem. (2)

(b) Develop with proof a recurrence for your strategy (3) efficient cost.

(c) Prove that a naive implementation of the (a) recurrence program will result in exponential time. (3)

(d) Develop a column-wise memoing based code and prove it based code. (3)

(e) Develop the exact complexity of (d). (4)

(2) (a) Define Partition ADT. (Open statement of problem) (2)

(b) Write pseudocodes for ADT operations in an implicit data structure representing an Partition. (You must include the Adjacent Lists data structure first) Use weighted tree merging and ancestor chasing during your code. (Do partial merge) (2+2+3)

(c) What is the worst case complexity for n ADT operations. (Prove your claim). (2)

(d) Construct a worst case example for (c). (3)

(3) Let $G = (V, E)$ be a weighted graph with 1-to vertices and edges are weighted. (u,v) and (u,v) denote that $w(u) > 0 \Rightarrow u \in V$ and $w(e) > 0 \Rightarrow e \in E$. Define the wt of a path is defined as the sum of its weights of all the vertices and edges present in the path. Show how you would modify Dijkstra algorithm to find shortest path with cost of path defined as above. Prove the correctness of your algorithm. (10 partial credit) (5)

(4) True or false? Justify your answer. (no partial credit)

i) If a connected graph with $|V|-1$ edges has a unique maximal edge e, then e can not be a part of a MST.

ii) If the lightest edge in a graph is unique then it must be a part of every MST.

iii) Prime Algorithm works even if there are negative edges. (2+2+2)

(5) Solve for $n = 2^k$ the following recurrences

$T(n) = 2T(n/2) + n \log n$. (3)

$T(n) = 2T(\frac{n}{2}) + \sqrt{n} \log n$. (3)

3 hrs — 4th Dec — 9-12 AM.

Sol Exam:

1(a) Given n matrices M_i, $1 \le i \le n$. compute $M = M_1 \times M_2 \times \cdots \times M_n$ with minimum number of multiplications. The dimension of M_i may be taken as $d_{i-1} \times d_i$.

1(b) Let C_{ij} be the optimum number of multiplications needed to evaluate the chain $M_i \times M_{i+1} \cdots \times M_j$, $1 \le i \le j \le n$.

$$C_{ij} = \min_{i \le k < j} \{ C_{ik} + C_{k+1j} + d_{i-1} d_k d_j \}$$

Proof is by usual cut & paste argument.

$C_{ii} = 0$

1(c) discussed in the class class. As in Cormen, 3rd ed, page 386

1(d) For $j = 1$ to n.
  Compute elements in column j.
  $((1,j-1), (2,j-1), \ldots (j-1,j-1))$.

That is,

For $j = 1$ to n
  For $i = j-1$ to 1.
    Compute C_{ij}.

Compute C_{ij}.
  $C_{ij} = \infty$
  For $k = i$ to $j-1$
    $t = C_{ik} + C_{k+1j} + d_{i-1} d_k d_j$.
    if $t < C_{ij}$
      $C_{ij} = t$.

Return C_{ij}.

No. of multiplications to compute $C_{ij} = 2(j-i)$.

Complexity is

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

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

$$= \sum_{j=1}^{n} j^2 - j.$$

$$= \frac{n(n+1)(2n+1)}{6} - \frac{n(n+1)}{2}$$

$$= \frac{n(n+1)}{6} [2n+1-3]$$

$$= \frac{n(n+1)(n-1)}{3}.$$

2. Let $S = \{1, 2, \ldots, n\}$ and $S_i \subseteq S$, $1 \le i \le k \le n$. Let $S_i \cap S_j = \phi$, $i \ne j$, and $\bigcup_{i=1}^{k} S_i = S$. Then, the collection of sets $(S_1, S_2, \ldots, S_k)$ is called a k-partition of S. On a given partition, we can perform two operations Union and Find.

Find(x), for an $x \in S$, returns the name of the set containing x. At the time of calling find. Union(A, B), where A & B are two sets of the partition, removes A & B from the collection and includes $A \cup B$ to the collection. (Thus a k-partition becomes a $(k-1)$ partition after performing a union operation).

2(b) An implicitly weighted tree is in the class class. Complexity $O(n \log n)$.

2(d) Any worst case example is fine.

4. $T(2^k) = 2T(2^{k-1}) + k \cdot 2^k$

$$= 2\left[2T(2^{k-2}) + (k-1) \cdot 2^{k-1}\right] + k \cdot 2^k$$

$$= 2^2 T(2^{k-2}) + (k-1) \cdot 2^k + k \cdot 2^k.$$

$$= 2^2 \left[2T(2^{k-3}) + (k-2) \cdot 2^{k-2}\right] + \cdots$$

$$= 2^3 T(2^{k-3}) + (k-2) \cdot 2^k + (k-1) \cdot 2^k + k \cdot 2^k$$

$$\vdots$$

$$= 2^k T(1) + (1) \cdot 2^k + 2 \cdot 2^k + \cdots + k \cdot 2^k$$

$$= 0 + 2^k (1 + 2 + \cdots + k)$$

$$= 2^k \frac{(k)(k+1)}{2} \quad \| \; k = \log n$$

$$= \Theta(n \cdot \log^2 n) \quad \| \; 2^k = n$$

Algebra is same

It will be $2^k \left(\frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{k}\right)$

$\sim n \log k \quad \| \; H(k) = \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{k}$

$\sim n \log \log n. \quad = \log k$

3(b) i)

NO. Every MST will have (c,d). That is the only edge connected d.

ii). Let $e = (v,u)$ be the unique min cost edge. If an MST T does not contain e, then $T + e$ will have cycle with other edges having strictly higher cost. Removing one of them any e', gives us $T' = T + e - e'$, with cost smaller than T. This contradicts minimality of T.

YES

iii Prime algorithm does for Min wt crossing edge and hence negative ~~edge~~ wt of edges does not affect the correctness.

Find(x).
  $y = x$;
  while $(y \ne p(y))$
    $y = p(y)$.
  Return (y).

Union(a, b).
  // a & b are the names of the roots.
  // a & b are at the roots of the respective trees.
  if $\text{Size}(a) \le \text{Size}(b)$
    $p(a) = b$
    $\text{Size}(b) = \text{Size}(b) + \text{Size}(a)$
    $\text{Size}(a) = 0$;
  else
    $p(b) = a$;
    $\text{Size}(a) = \text{Size}(a) + \text{Size}(b)$
    $\text{Size}(b) = 0$;
  // smaller tree is hooked to larger tree.