CSE 373 – Analysis of Algorithms **Midterm Exam**
Piyush Kumar **Oct 20th, 2003**

Name: _____ ID #: _____

INSTRUCTIONS:

- This is a closed book, closed notes exam.
- You may use either pen or pencil.
- Check to see that you have 10 pages including this cover and scratch pages.
- Look over all the problems before starting work.
- Think before you write.
- Good luck!!

Academic integrity is expected of all students at all times, whether in the presence or absence of members of the faculty.

Understanding this, I declare that I shall not give, use, or receive unauthorized aid in this examination. I have been warned that if I am caught cheating (either receiving or giving unauthorized aid) I will get a "Q" grade for this course, and a letter will be sent to the Committee on Academic Standing and Appeals (CASA) requesting that an academic dishonesty notation be placed on my transcript. Further action against me may also be taken.

Signature: _____

Problem	Score	Maximum
1		15
2		10
3		16
4		20
5		15
6		24
Total		100

Problem 1. (15) True or false? **To get credit you must justify your answer**. There is no credit for answers without correct justification.

(a) **T** **F** $f(n) = O(g(n))$ iff $2^{f(n)} = O(2^{g(n)})$.

(b) **T** **F** $2^{\log(n)} = \Omega(n^c)$, for constant $c > 1$ (not depending on n).

(c) **T** **F** Two $n \times n$ matrices can be multiplied in $O(n^{2.9999})$ time.

(d) **T** **F** $\log(n) = \Theta(n^{\log \log n / \log n})$.

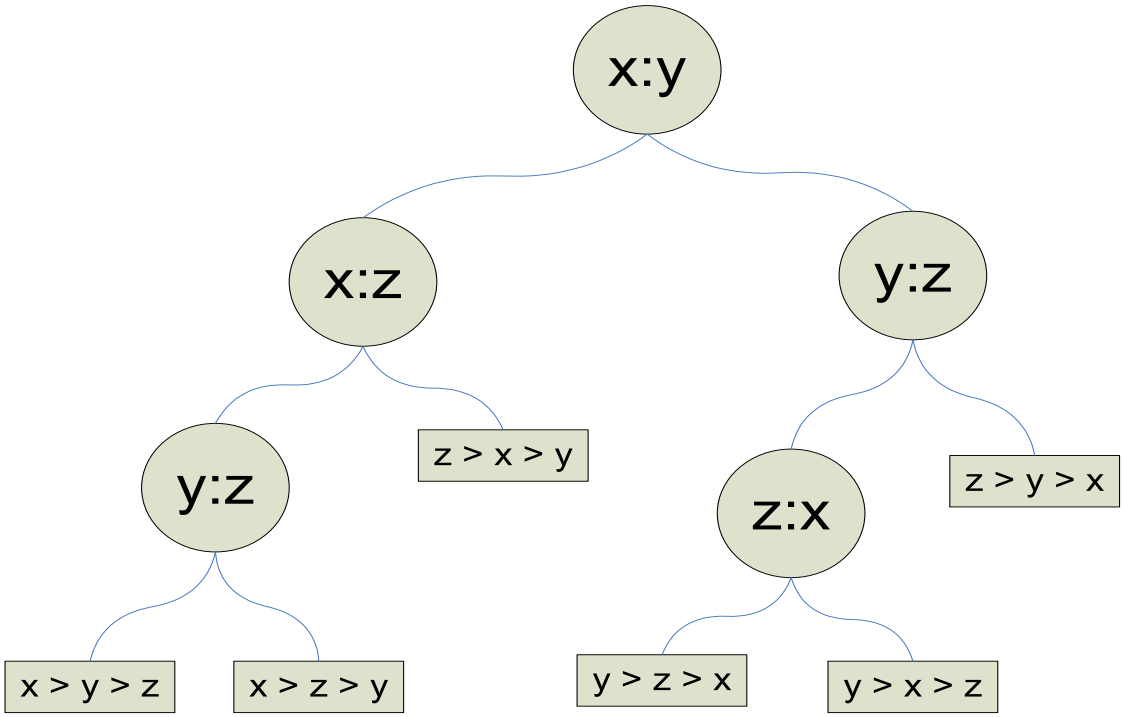
(e) **T** **F** Given a set of n points in the plane, one can determine the closest pair in $O(n \log n)$ time. (If you say true, write the recurrence.)

Problem 2. (10)
Suppose that we have an array $A[1 \ldots n]$ of unsorted real numbers . Elements may be replicated.

(a) Our objective is to find the element that appears **most frequently** in the array. How long does your algorithm take to run? _____

(b)What is the main idea in a few words._____

Problem 3. (16)



(a) What is the best case number of comparisons for the given comparison tree? ___

(b) What is the worst case number of comparisons? ___

(c) What is the expected number of comparisons needed for this comparison tree given that all leaves are equally likely (Dont worry if you get a fraction, note that average number of comparisons = expected number of comparisons in this case)? ___

(d) **T** **F** Can you sort 5 elements with 7 comparisons using a comparison based sorting algorithm? (Justify your answer)

Problem 4. (20)
(a) Suppose T is a multi-way search tree such that each internal node has at least three and at most five children. For what values of a and b is T a valid (a, b)-tree?

(b) Describe in detail how would you insert an item in an (a, b)-tree? What main operations do you require, show them using a picture.

Problem 5. (15)
Write the letter associated with the correct solution for each of the problems. To get credit you must justify your answer!
Note: A letter may appear multiple times or not at all.

____ $T(n) = T(n/2) + n, \quad T(3) = T(2) = T(1) = 1.$

____ $T(n) = 3T(n/3) + 1, \quad T(2) = T(1) = 1.$

____ $T(n) = T(n-1) + n, \quad T(1) = 1.$

Possible Solutions :

a) $\Theta(\log \log n)$

b) $\Theta(\log n)$

c) $\Theta(\sqrt{n})$

d) $\Theta(n)$

e) $\Theta(n \log \log n)$

f) $\Theta(n \log n)$

g) $\Theta(n \log^3 n)$

h) $\Theta(n^2)$

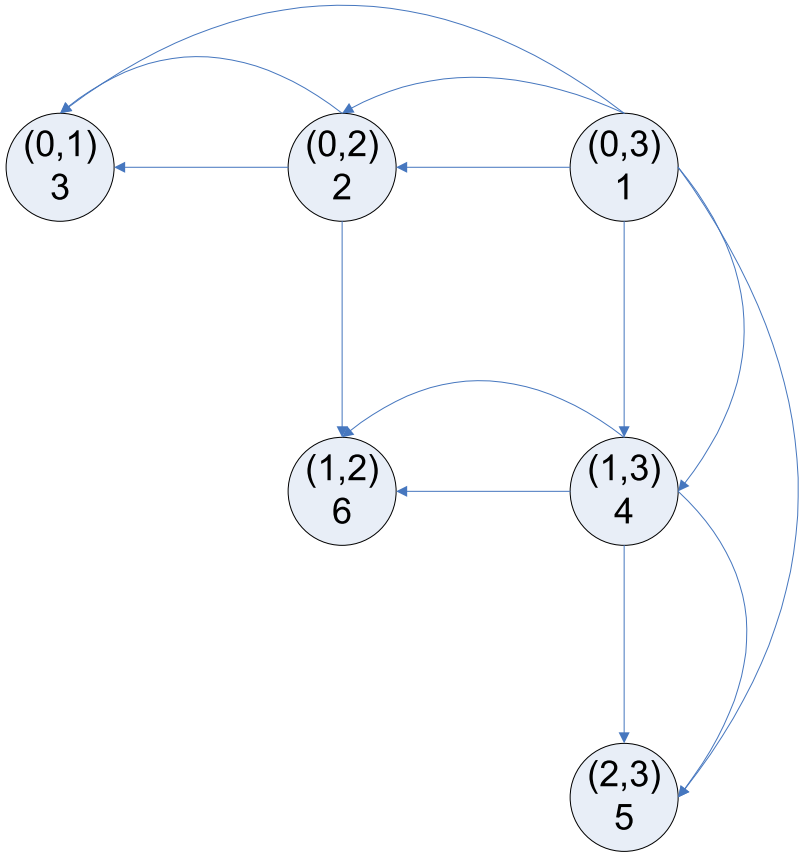
i) $\Theta(n^3)$

j) $\Theta(n^3 \log n)$

k) none of the above

Working space

Problem 6. (24)
In this problem we are going to consider the subproblem graph for ordered matrix multiplication. Let A_1, A_2, A_3 be matrices of dimensions $d_0 \times d_1, d_1 \times d_2, d_2 \times d_3$ respectively.



(a) **T** **F** For the above graph, is $3, 6, 5, 2, 4, 1$ a depth first search traversal of the subproblem graph?

(b) **T** **F** For the above graph, is $3, 6, 2, 5, 4, 1$ a depth first search traversal of the subproblem graph?

(c) Give two traversals of the sub problem graph for calculating the order for multiplying A_1, A_2, A_3 such that a dynamic programming algorithm can use the traversal to calculate its subproblems. For instance a dynamic programming algorithm can use the traversal, $3, 6, 5, 2, 4, 1$ (This is the traversal we used in the iterative version of the DP for the solution to this problem. You need to give two other traversals excluding this traversal.)

(d) Fill in the transitive closure of the adjacency matrix representation of the subproblem graph given.

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ -- & -- & 0 & -- & -- & -- \\ -- & -- & -- & 0 & -- & -- \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(e) Let us suppose now that you have n matrices (The figure shows you the subproblem graph for $n = 3$). How fast can you compute the transitive closure of the subproblem graph for this problem? Give your solution in Order notation.

(f) What problem in class did we do that is close to the problem of computing the transitive closure of a graph?

Working space

Problem 1.
In this problem we use *modular arithmetic*. We say that for integer x and positive interger q

$$x \bmod q$$

is the remainder after dividing x by q. An (inefficient) way to calculate the mod is: if x is positive keep subtracting q from x until it is first less than q. If x is negative keep adding q until it becomes nonnegative. (e.g. $-3 \bmod 7 = 4$, $209 \bmod 40 = 9$.) We can assume that mod can be calculated in constant time. Note that this is just an explanation but not the important part of the problem since taking the mod can be done in constant time.

In the RSA cryptosystem the main way to encrypt and decrypt a message M is described as follows. The message is a sequence of bits. We view this bits as a large number and raise it to a large power k. (Typically k is between 100 and 1000 bits long.) All of the computation is done modulo a large number N. E.g. we compute

$$M^k \bmod N.$$

(a) What is the running time of the algorithm? _____

(b) To get credit from (a) you *must* write a brief justification of the idea.

(Remember to sign the cover page!)

Problem 2.
Recall that in the edit distance problem $E[i, j]$ is the edit distance between $A[1 \ldots i]$ and $B[1 \ldots j]$. Suppose that we change the costs of edits, as follows. It costs "2" to insert or delete a character but only "1" to change a character.
(a) Fill in the table $E[0 \ldots 5][0 \ldots 3]$ below assuming these new costs. Some of the spaces are already filled in.

- $A[1 \ldots 5]$ =ababa
- $B[1 \ldots 3]$ =baa

		a	b	a	b	a
	0	2	4	6	8	10
b	2	1	2	4	6	8
a	4					
a	6					

(b) Please write the formula that generates the above table. Just write the recurrence relation and *nothing else*.

Problem 3. True or false? **To get credit you must justify your answer**. There is no credit for answers without correct justification.

(a) **T** **F** $2^{\log(n)} = \Omega(n^c)$, for constant $c > 1$ (not depending on n).

(b) **T** **F** For monotonically increasing functions $f(n),\ g(n) > 1$, if $f(n) = \Theta(g(n))$ then $n^{f(n)} = \Theta(n^{g(n)})$.

(c) **T** **F** $2^{2n} = O(3^n)$

(d) **T** **F** $\log(n) = \Theta(n^{\log\log n/\log n})$.

(e) **T** **F** $n^{1.2} = \Omega(n \log^2 n)$

Problem 4.
Suppose that we have an array $A[1 \ldots n]$ of unsorted real numbers . Elements may be replicated.

(a) Our objective is to find the element that appears **most frequently** in the array. How long does your algorithm take to run? _____

(b)What is the main idea in a few words._____

Problem 5. (16)
Write the letter associated with the correct solution for each of the problems. To get credit you must justify your answer!
Note: A letter may appear multiple times or not at all.

_____ $T(n) = 3T(n/2) + n^3, \quad T(3) = T(2) = T(1) = 1.$

_____ $T(n) = 3T(n/3) + n, \quad T(2) = T(1) = 1.$

_____ $T(n) = T(n-1) + n, \quad T(1) = 1.$

Possible Solutions :

a) $\Theta(\log \log n)$

b) $\Theta(\log n)$

c) $\Theta(\sqrt{n})$

d) $\Theta(n)$

e) $\Theta(n \log \log n)$

f) $\Theta(n \log n)$

g) $\Theta(n \log^3 n)$

h) $\Theta(n^2)$

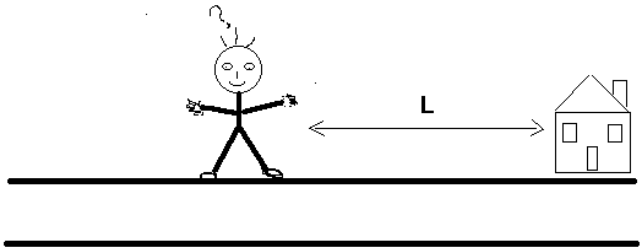
i) $\Theta(n^3)$

j) $\Theta(n^3 \log n)$

k) none of the above

Working space

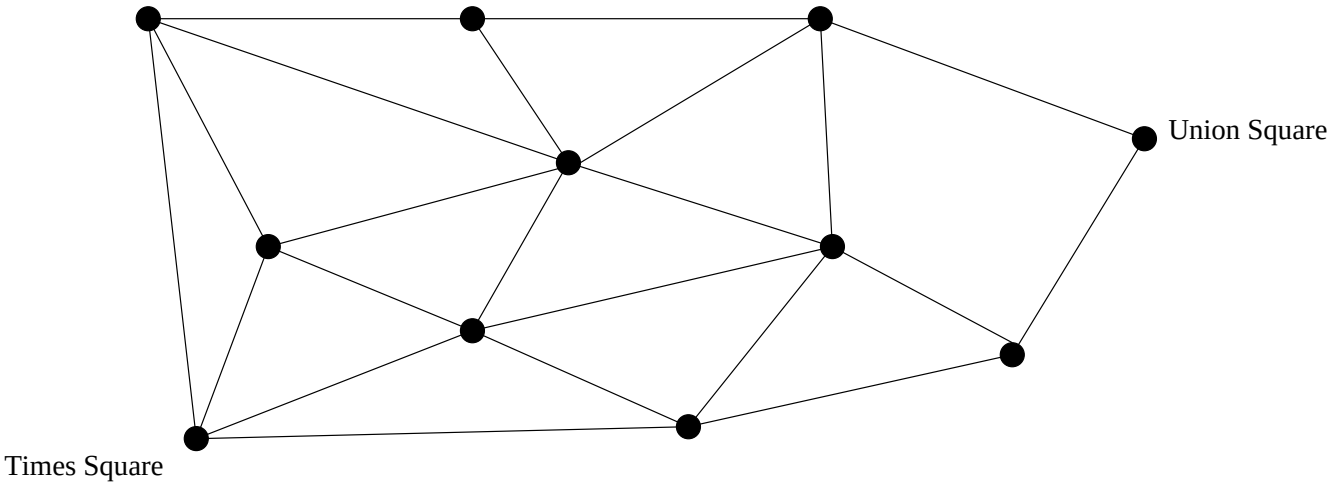
Problem 6. (16)
You are standing on a street looking for a particular house. But you do not know if you should go east or west. The house is at distance L from you, but you do not know what L is.



(a) Give an algorithm to find the house. The algorithm should run in time $O(L)$.
Hint: This should remind you of a homework problem.

(b) Draw a picture to illustrate your idea. (If you have a very good picture, then you do not even need to write words for part (a).)

Problem 7. (16)
In this problem we are going to consider algorithms on city blocks. Nodes correspond to intersections and edges correspond to streets.



(a) Suppose we want to know the smallest number of intersections between times square and union square. What algorithm or algorithms should we use and slightly modify.

(b) What is the running time. To get credit you *must* explain why or draw a picture.

Problem 1. (10) Write the letter associated with the correct solution for each of the problems. To get credit you must justify your answer!
Note: A letter may appear multiple times or not at all.

_____ $T(n) = 2T(n/2) + \log n, \quad T(1) = 1.$

_____ $T(n) = T(n/2) + n, \quad T(1) = 1.$

Possible Solutions :

a) $\Theta(\log\log\log n)$

b) $\Theta(\log\log n)$

c) $\Theta(\log n)$

d) $\Theta(\log^2 n)$

e) $\Theta(\log^3 n)$

f) $\Theta(\sqrt{n})$

g) $\Theta(n)$

h) $\Theta(n\log\log n)$

i) $\Theta(n\log n)$

j) $\Theta(n\log^2 n)$

k) $\Theta(n^2)$

l) $\Theta(n^3)$

m) $\Theta(n^3\log n)$

n) $\Omega(1.1^n)$

o) none of the above

Problem 2.(8)
In the following problems you should fill in the asymptotic complexities of the operations of the data structures or algorithms.

(1) In the UNION-FIND data structure:
Cost of FIND in array-based structure: _____.

Cost of FIND in tree-based structure: _____.

Cost of FIND with path compression: _____.

(2) In the BALANCED SEARCH TREE data structure:
insert(x) _____, search(x) _____
print out all elements in sorted order_____.

(3) In the SKIP LIST data structure:
insert(x) _____, search(x) _____
print out all elements in sorted order_____.

(4) In the HASH TABLE data structure:
insert(x) _____, search(x) _____
print out all elements in sorted order_____.

(5) In the **adjacency matrix** representation of a graph $G(V, E)$, where V is the set of nodes and E is the set of edges:
BFS on G _____, DFS on G (x) _____

(6) In the **adjacency list** representation of a graph $G(V, E)$, where V is the set of nodes and E is the set of edges:
BFS on G _____, DFS on G (x) _____

Problem 3.(8)
Suppose that the edge weights of a graph $G(V, E)$ are $\sqrt{1}, \sqrt{2}, \sqrt{3}, \ldots, \sqrt{n}$, where $n = |E|$. How many Minimum Spanning Trees are there? Prove your answer is correct.

Problem 4. (25)
Provide a convincing explanation or a counter example. Circle the correct answer. In this question, as in all the remaining questions, you **must** show your work to get credit.
(a) Which of these changes when we *add a constant* to each edge?

1. shortest path changes stays the same

2. minimum spanning tree changes stays the same

3. traveling salesman tour changes stays the same

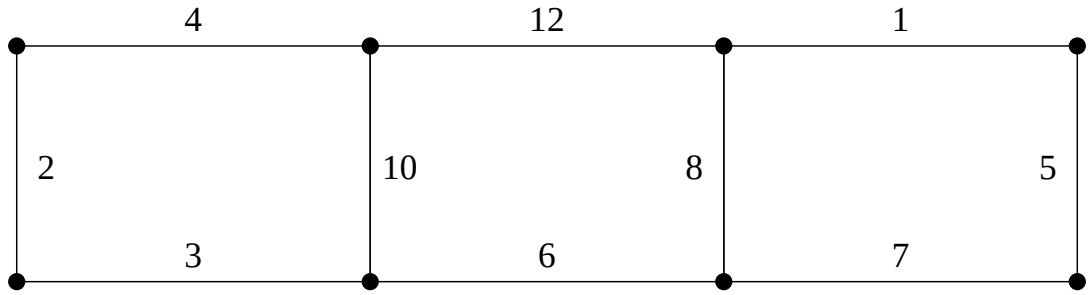
(b) Which of these changes when we *multiply each edge* by a constant?

1. shortest path changes stays the same

2. minimum spanning tree changes stays the same

3. traveling salesman tour changes stays the same

Problem 5. (8)
In the *bottleneck spanning tree problem* we define the *cost* of a spanning tree to be the weight of the *heaviest* edge. We want to find the spanning tree of a graph G with the minimum cost. That is, we want to minimize $\max_{e \in E}\{weight(e)\}$. We call such a spanning tree a *Bottleneck Spanning Tree*.
(a) Draw a Bottleneck Spanning Tree on the graph below:



(b) **T** **F** A Minimum Spanning Tree is a Bottleneck Spanning Tree.

Prove or provide a counter example.

(c) **T** **F** A Bottleneck Spanning Tree is a Minimum Spanning Tree.

Prove or provide a counter example.

(d) Provide a SIMPLE polynomial-time algorithm to find a Bottleneck Spanning Tree. (Just one sentence please!!)

(e) Given a graph $G(V, E)$ and a positive integer K, Provide an algorithm to find out whether G has a spanning tree with cost less or equal to K in $O(|V| + |E|)$ time.

Problem 6. (10)

(a) Design a data structure implementating the following 4 operations:

(1) Print out all emlements in the data structure in order ($O(n)$ time).

(2) Insert an element ($O(\log n)$ time).

(3) Delete an element ($O(\log n)$ time).

(4) Return the max element **ever** inserted (even if it was subsequently deleted) ($O(1)$ time).

(b) Suppose you are going to design a data structure to maintain records of employees in your company. Each employee record has two fields: name, salary. You need to design a data structure implementating the following operations: For simplicity, you can assume all names are dinstinct and all salaries are distinct.

(1) Print out all employee records ordered by their names ($O(n)$ time).

(2) Hire(n, s): insert a new employee with name n and salary s ($O(\log n) time$).

(3) Fire(n): Delete an employee with name n ($O(\log n) time$).

(4) Salary(n): return the salary of employee with name n ($O(\log n) time$).

(5) TopDog: Return the employee currently being hired with the highest salary. ($O(1)$ time).

What two data structures from class are used to solve the problem?

_____ _____

Draw a picture of your data structure and explain your idea.

Problem 7.(10)
In class, we studied the UNION-FIND data structure. Now we will consider a related data structure called the SPLIT-UNION-FIND data structure. In SPLIT-UNION-FIND we are given a universe of *integers* $\{1, 2, \ldots, n\}$. Our goal, is to maintain disjoint sets. However, here the sets must consist of *contiguous* integers. Each set $\mathcal{S}$ has a *representative*, which is the *smallest* element in $\mathcal{S}$. Thus, at all times, if FIND(i) =FIND(k), then for all j such that $i \leq j \leq k$, FIND(i) = FIND(j) = FIND(k).

As an example, we might have a universe $\{1, 2, \ldots, 8\}$, and have sets

$$\boxed{1\ 2\ 3}\quad \boxed{4}\quad \boxed{5\ 6\ 7}\quad \boxed{8}.$$

In this example, FIND(5) = FIND(6) = FIND(7) = 5.

We are interested in performing the following operations:

FIND(k). Returns the smallest element in the set $\mathcal{S}$ containing k (the representative).

SPLIT(k). Given a element k, we split the set $\mathcal{S}$ containing k into two pieces. The first piece contains all the elements in $\mathcal{S}$ that are $\leq k$. The second piece contains all the elements in $\mathcal{S}$ that are $> k$.

UNION(k). Merge the set containing k, with the next largest set (on the right on the number line).

Thus, in the example, if we perform SPLIT(2), we obtain,

$$\boxed{1\ 2}\quad \boxed{3}\quad \boxed{4}\quad \boxed{5\ 6\ 7}\quad \boxed{8}.$$

If we next perform UNION(4), we obtain

$$\boxed{1\ 2}\quad \boxed{3}\quad \boxed{4\ 5\ 6\ 7}\quad \boxed{8}.$$

(A) For the best implementation that you can find of this data structure, what is the cost of the following operations?

FIND: _____. SPLIT: _____. UNION: _____.

(B) What is the basic idea behind your data structure in part (A)? (Only one sentence!)

Problem 8.(10)
Recall the Minimum Spanning Tree approximation to the Traveling Salesman Problem. For this problem suppose that you have a graph G that does **NOT** obey the triangle inequality.
(a) Is the minimum spanning tree of G still a lower bound on the optimal traveling salesman tour? Why or why not?

(b) Suppose that you want to find a tour that visits each vertex of G at least once. Devise a $2 * OPT$ approximation algorithm that uses the minumum spanning tree.

(c) Now suppose that you want to find a tour that visits each vertex of G once. Can you still use your algorithm from (b)? Why or why not?

Problem 9.(10)
Suppose we have a set of jobs to be scheduled on p processors. Each job takes one operation on one processor to finish. The dependency graph of the jobs forms a balanced binary tree of height h.

(a) What is the order of the amount of *total work*? _____
(By "total work" I mean the number of operations.)

(b) What is the order of the length of the *critical path*? _____

(c) What strategy should be used for scheduling on p processors? _____

(d) How long is the schedule? _____

Problem 9.(10)
You are designing a web server for google. Each request to the web server is a string. Some of these requests have already been made recently and the web server response is still in main memory. Other responses are on disk.
You need to write a program to determine which requests are stored in main memory and which are on disk as fast as possible.
What is the main idea? What data structure will solve to problem?

Please provide a few details.

Problem 9.(10)
We are trying to organize a collection of N books by similarity. We want to place the books on a bookshelf so that any pair of books next to each other on the shelf are as similar as possible. For each pair of books (i, j), the measure of their difference is already determined, eg, is some know quantity d_{ij}.

We want to order the books on the shelf to minimize the sum of the differences of all neighbors, that is,

$$\sum_{\text{book}_{i,j}\text{are neighbors on shelf}} d_{ij}.$$

(1) What problem in CS models this situation?

(2) Is the problem NP-hard or polynomial time?