Fall 2009, Final Exam. Data Structures and Algorithms

Name:
Section:
Email:

21st December, 2009 (Winter Solstice)

This is an open book, open notebook exam. Answer all twelve questions. Each question is worth 10 points. You have 180 minutes to complete the exam.

Happy Holidays and Have a Great New Year.
1. (a) \text{\textbf{Max SAT} \ is \ \textit{NP}-complete.}

\textit{Given an assignment,}
\textit{and see if at least \( g \) clauses are satisfied.}

\textit{CNF SAT \ is \ \textit{polynomial time} transformable to \textit{Max SAT}.}

\textit{Given: CNF SAT problem \( \Phi \).}
\textit{? : Does \( \Phi \) \text{ have an assignment \( \Phi^* \) that}
\text{satifies all clauses \( \Phi \) ?}}

\textit{Instance for Max SAT:}
\textit{Given: \( P, \Phi^* (g=0) \).
? : Does \( \Phi^* \) \text{ \text{ satifies } \( \Phi \) ?}\}
\textit{Assignment \text{ satifies } \( \Phi \) \text{ in Max SAT \( \Phi \).}}
\textit{As \text{ from CNF is satifies.}}

(b) \text{ Take subset \( g \) of vertices and check \text{ whether}
\text{ the given \( \Phi \) satisfies a \text{ clique of } \( g \). \text{ If yes, report clique of } \( g \) \text{ else}
\text{ take subset \( g \) of vertices and check \text{ whether}
\text{ the given \( \Phi \) \text{ satisfies a smaller clique of } \( g \)}}
\text{ \text{ Clique is } \{ \Phi \} \times \text{.} \}
\text{ Complexity is } O(n^4).
2. Local Search

Problem 9.6 (DG)

In the Minimum Steiner Tree Problem, the input consists of: a graph \( G = (V, E) \) with distances \( d_{uv} \) between all pairs of nodes (distances satisfy metric property); and a distinguishable set of terminal nodes \( V' \subseteq V \). The goal is to find an efficient, ratio-2 approximation algorithm for finding a minimum Steiner tree that includes vertices \( V' \). This tree may or may not include vertices in \( V \). (Hint: Construct all pairs of shortest distances between vertices in \( V' \) and what can you say about these distances? Then construct a Minimum Spanning Tree and select edges from the original graph.)

1. Find a minimum spanning tree \( T' \) from the induced subgraph \( G' \) on \( V' \).

2. Chain the MST in at most twice the cost of Steiner tree.

Proof: Let \( T \) be a Steiner tree.

Visually, the Steiner tree is a minimum spanning tree. The cost of the MST \( T' \) is at most twice the cost of the Steiner tree.
3. Graphs/DFS/Path

Consider the following graph with 10 vertices and 15 edges.

10 points
(a) Draw a BFS tree starting at vertex 1.
(b) Draw a DFS tree starting at vertex 1.
(c) What is the longest path from vertex 1 and vertex 10.
(d) Draw two vertex paths from vertex 1 to vertex 2.
Shortest Path Algorithms/Spanning Tree Consider 8 of 3-bit binary numbers from 000 to 111. Treat them as vertices of an undirected graph. Two vertices are adjacent if they differ in exactly one bit position. For example vertex labeled 001 is adjacent to 101, 011, 000.

(a) Draw a graph of this 8 vertex undirected graph.

(b) Draw a cycle of length 8 from vertex labeled 000 to vertex labeled 000.
5. Linear Programming and Matching

(a) Do problem 7.3 (just the LP formulation): A cargo plane can carry a maximum weight of 100 tons and maximum volume of 60 cubic meters. There are three materials to be transported, and the cargo company may choose to carry any element of each, up to the maximum available limits given below:

- Material 1 has density 2 tons/cubic meter, maximum available amount is 40 cubic meter and revenue is $1000 per cubic meter.
- Material 2 has density 4 tons/cubic meter, maximum available amount is 30 cubic meter and revenue is $1200 per cubic meter.
- Material 3 has density 3 tons/cubic meter, maximum available amount is 20 cubic meter and revenue is $1200 per cubic meter.

Write a linear program that optimizes revenue within the constraints:

\[
\begin{align*}
\text{maximize: } & 1000x_1 + 1200x_2 + 1200x_3 \\
\text{subject to: } & 2x_1 + x_2 + 3x_3 \leq 100 \\
& x_1 + x_2 + x_3 \leq 60 \\
& x_1 \leq 40 \\
& x_2 \leq 30 \\
& x_3 \leq 20 \\
& x_1, x_2, x_3 \geq 0
\end{align*}
\]

(b) Draw a tree with 6 vertices and has a max matching with three edges.

\[
\begin{align*}
\{1, 2, 3, 4, 5, 6\} & \text{ is a matching}
\end{align*}
\]
(a) Consider the directed acyclic graph $G$ in the above figure. How many topological orderings does it have? (What are all the possible labeling of the vertices that preserve topological order?)

(b) Describe an algorithm to find the largest spanning tree in a given undirected weighted graph.

$$4 \text{ possibilities}$$

(b) In Kruskal’s algorithm, sort all edges in non-decreasing order of weight. Select the longest weight edge at any step that can be included in the max.

Spruce.
7. **Network Flows** Consider the following network (the numbers are edge capacities). What is the maximum flow from S to T and what is the minimum cut?
Dynamic Programming

Given two strings \( x = x_1 x_2 \ldots x_n \) and \( y = y_1 y_2 \ldots y_m \), we wish to find the length of their longest common substring, that is, the largest \( k \) for which there are indices \( i \) and \( j \) with \( x_i, x_{i+1}, \ldots, x_{i+k-1} = y_j, y_{j+1}, \ldots, y_{j+k-1} \). Show how to do this in time \( O(nm) \).

\[
L(x_i, y_j) = \left\{ \begin{array}{ll}
L(x_{i-1}, y_{j-1}) + 1 & : x_i = y_j \\
0 & : \text{otherwise}
\end{array} \right.
\]

\[
s(i,j) = \text{longest common substring of } \ x_i \ldots x_{i+k-1} \text{ and } y_j \ldots y_{j+k-1}
\]

\[
s(i,j) = \max_{i' < i, j' < j} s(i', j') + 1 \\
= \begin{cases} 
0, & \text{for } i = 0, j = 0, \\
1, & \text{for } i = 0, j > 0, \\
1, & \text{for } i > 0, j = 0, \\
\max_{k} s[i-k, j] + 1, & \text{otherwise}.
\end{cases}
\]

For \( i = 0 \) to \( n \)
\[
s[i, 0] = 0
\]

For \( j = 0 \) to \( m \)
\[
s[0, j] = 0
\]

For \( i = 0 \) to \( n \)
\[
\text{for } j = 0 \text{ to } m
\]
\[
\text{if } x(i) = y(j)
\]
\[
s[i,j] = s[i-1, j-1] + 1
\]
\[
\text{else}
\]
\[
s[i,j] = 0
\]

return \( \max_{i,j} s[i,j] \).
9. **Huffman** Suppose that the symbols \(a, b, c, d, e\) occur with frequencies of \(1/2, 1/4, 1/8, 1/16, 1/16\) respectively.

(a) What is the Huffman encoding of this alphabet?

(b) If this encoding is applied to a file consisting of 1,000,000 characters with the given frequencies, what is the length of the encoded file in bits?
(a) Give a linear time algorithm that takes input as a tree \( T \) and determines a two-coloring of the vertices, such that each vertex is colored with one of two colors and no two adjacent vertices are not colored with the same color.

\[
\text{DFS on } T: \quad \text{Solve the node or DFS number of the node } \mod 2.
\]

(b) Solve the equation for integers \( x \) and \( y \) such that \( 13 \times x + 8 \times y = 1 \), \( \gcd(13,8) = 1 \).

\[
\begin{align*}
13 &= 1 \times 8 + 5 \\
8 &= 1 \times 5 + 3 \\
5 &= 1 \times 3 + 2 \\
3 &= 1 \times 2 + 1 \\
2 &= 2 \times 1 + 0
\end{align*}
\]
11. Algorithm Design

An array \( A[1 \ldots n] \) (random access) is said to have a majority element if more than half of its elements are the same. Given an array, design an efficient algorithm to tell whether there is a majority element and, if so, find that element. The only question/operation that you are allowed to do is \( A[j] = A[j] \) and this operation takes constant time. A pseudocode description will suffice. Analyze your algorithm.

\[
\text{potential max} \leftarrow 99999
\]

\[
\text{for } i = 1 \text{ to } n
\]

\[
\begin{cases} 
\text{if potential max} = 99999 \\
\text{potential max} \leftarrow A[i] \\
\text{else} \\
\text{if potential max} = A[i] \\
\quad \text{count} + 1 \\
\text{else} \\
\quad \text{count} - 1 \\
\quad \text{if count} = 0 \\
\quad \text{potential max} \leftarrow 99999 \\
\end{cases}
\]

\[
\text{if potential max} = 99999
\]

\[
\text{for } i = 1 \text{ to } n
\]

\[
\begin{cases} 
\text{count} \leftarrow \text{number of times potential max} \\
\text{if } A[i] = \text{potential max} \\
\quad \text{count} + 1 \\
\end{cases}
\]

\[
\text{if count} > n/2 \\
\quad \text{print potential max} \\
\text{else print no majority element}
\]

\( O(n) \)
12. Multiplication and Divide and Conquer

(a) Describe an efficient algorithm to multiply two n digit complex integers (Hint: Use divide and conquer strategy). (A + iB) * (C + iD) = (AC - BD) + i(AD + BC).

\[(A + iB) \times (C + iD) = M_1\]
\[(A + iB) = M_2\]
\[(B + iD) = M_3\]
\[(M_2 - M_3) + i(M_1 - M_2 - M_3)\]

(b) Find the value of \(T(8)\) and then solve the following recurrence equation for general \(n = 2^k\).

\[T(n) = 3T(n/2)\]

and \(T(1) = 1\).

\(n = 2^k\)

\[T(2^k) = 3T(2^{k-1})\]

\(Q(k) = 3Q(k-1)\)

\(Q(0) = 1\)

\(Q(k) = 3^k \cdot 1 = 3^k\)

\[T(2^k) = 3^{k+1}\]

\[T(n) = 3^{\log_2 n}\]

\[= n^{\log_2 3}\]

\[= 0 (n^{\log_2 3})\]