
Homework 4 – Due Friday, September 26, 2008

Please refer to the general information handout for the full homework policy and options.

Reminders

- Your solutions are due before the lecture. Late homework will not be accepted.
- Collaboration is permitted, but you must write the solutions *by yourself without assistance*, and be ready to explain them orally to a member of the course staff if asked. You must also identify your collaborators. *Getting solutions from outside sources such as the Web or students not enrolled in the class is strictly forbidden.*
- To facilitate grading, please write down your solution to each problem on a separate sheet of paper. Make sure to include all identifying information and your collaborators on each sheet. Your solutions to different problems will be graded separately, possibly by different people, and returned to you independently of each other.
- For problems that require you to provide an algorithm, you must give a precise description of the algorithm, **together with a proof of correctness** and an analysis of its running time. You may use algorithms from class as subroutines. You may also use any facts that we proved in class or from the book.

Exercises These should not be handed in, but the material they cover may appear on exams:

1. Exercises in Chapter 4.
2. (MST with negative-weight edges) Give a $O(m \log n)$ algorithm which takes as input a connected, undirected graph $G = (V, E)$ and distinct weights w , *some of which may be negative*, and outputs a minimum spanning subgraph $F \subset E$, i.e. it outputs a set of edges F such that (V, F) is connected and $\sum_{e \in F} w_e$ is minimized. Note that F need not be a tree.
3. Suppose G is a connected, undirected graph in which all edge weights w_e are positive and distinct. Consider the graph G' with the same vertices and edges as G , but with edge weights $w'_e = w_e^2 + 6$. Prove or disprove: For every pair of vertices u and v , the shortest path from u to v is the same in G as in G' .
4. Suppose G and G' are as above. Prove or disprove: The minimum spanning tree of G' is the same as the minimum spanning tree of G .

Problems to be handed in

<p>Page limits: The answer to each problem should fit in 2 pages (or one double-sided sheet) of paper. Longer answers will be penalized.</p>
--

1. **(Recursion-Tree Method)** Use the recursion-tree method to solve the following recurrences. Express your answer using Θ -notation. Note: It may help to review sums of geometric and arithmetic progressions (e.g. sums like $\sum_{i=1}^n r_i$ or $\sum_{i=1}^n i$.)
 - (a) $T(n) = 7T(n/2) + cn$, where $c \geq 0$ is a constant.
 - (b) $T(n) = T(n-1) + c \log^2(n)$, where $c \geq 0$ is a constant.
 - (c) $T(n) = T(n/5) + 4T(4n/5) + cn$, where $c \geq 0$ is a constant.

2. **(MST)**
 - (a) Suppose G is a connected, undirected graph in which all edge weights w_e are positive but not necessarily distinct. Let $S \subset V$ be a cut in the graph, and let E_S be the set of least-weight edges with exactly one endpoint in S (there may be several such edges since edge weights are not distinct). Prove or disprove: for every edge e in E_S , there exists a minimum spanning tree T for G containing e .
 - (b) KT, Chapter 4, Problem 19 (bottlenecks).

3. **(Space-Efficient Algorithms for the MST)** You are designing a MST algorithm that runs on a machine with a large input but very small workspace. The input to your algorithm is an undirected graph presented in the following form: an integer $n = |V|$, followed by a list of m edges along with their weights, that is, a list of triples (u, v, w_{uv}) , where $u, v \in \{1, \dots, n\}$ and $w_{uv} > 0$.
 Give an algorithm which makes a single pass through the list of edges, using only $O(n)$ space, and outputs a minimum spanning tree at the end of its pass. Your algorithm should work no matter what order the edges are listed in.
 (Hint: as you pass through the input, maintain a tree; as each new edge is considered, decide whether or not to add it to your tree and, if necessary, which edges from the current tree to discard).
 You may assume that the graph is connected and that all edge weights are distinct. Remember to prove the correctness of your algorithm and discuss its running time.

4. *(Extra credit) Give an algorithm for the previous problem that uses $O(n \log n)$ space and $O((m+n) \log n)$ total time.

Note on the motivation for this model: Data sets are often too large to fit into main memory. For example, AT&T's phone call graph has millions of vertices (users) and trillions of edges (phone calls). Storing a spanning tree for such a graph would require only megabytes of space, whereas storing all edge of the graph requires terabytes! The algorithms you are asked to design here use a small amount of main memory to process a large secondary memory via a single pass. Alternatively, you can think of edges arriving in real time (say, as phone calls are made), and the system keeping track only of the MST of the graph rather than all its edges.