Single Source Shortest Paths

11-13-2003

Opening Discussion

- What is a spanning tree? What does it mean to say we have a minimum spanning tree? What is the rough algorithm to find such a thing?
- Do you have any questions about the assignment?
- Why didn't I make the graph generic? Let's look at some sample graph problems to help illustrate the reason.

Kruskal's Algorithm

- In this algorithm we always add the lowest weight edge in the graph that doesn't create a cycle. As a result, while the algorithm executes, we go through a process of connecting a forest of trees to produce a single tree.
- To make this fast we first sort the elements of E by weight and simply walk that in a loop then use fast structures for disjoint sets (Ch. 21).

Prim's Algorithm

- This algorithm starts with a particular node, and adds edges to a single tree, A, that is built. At each step the edge added is the smallest to connects the tree to a vertex not yet in the tree.
- To make it efficient we keep a min-priority queue of the "keys" of nodes where a key is the minimum edge connecting a node not in the tree to some node in the tree.

Shortest Paths

- You want to get from *u* to *v* and you want to know the shortest way to get there. In an unweighted graph this is simply the path constructed with a breadth first tree that is rooted at *u*.
- For a weighted tree this is slightly more complex.
- WLOG we can say that shortest paths do not contain cycles.

Shortest Path Trees

- If we have a single source, and want to find the shortest path to all other vertices, we can build a tree that represents those paths. This tree is only the shortest path tree at the end, during the processing it can contain branches that are not part of the final tree.
- We will use relaxation algorithms to do this. We keep in each node the minimum distance that has been found so far as we intelligently add edges to the tree. When we find a shorter path to a given node, we alter the parent as well.

Bellman-Ford Algorithm

- For this algorithm we initialize all the data (set d[u]=∞ and p[u]=nil). Then we run through all edges in the graph |V|-1 times and relax all of the destination nodes.
- When done we can detect negative cycles which prevent the algorithm from giving the proper result. This happens if there is an edge (u,v) in E and d[v]>d[u]+w(u,v).
- This takes O(V*E) time.

Minute Essay

- Now do you understand why graphs and their algorithms are hard to write in a generic way? If you didn't have the book, how would you try to go about finding minimum spanning trees or shortest paths? How would you improve on the shortest path algorithm discussed here?
- Quiz #5 will be next class.