Washington University in St. Louis

Engineering

Review Session

Jon Turner
Computer Science & Engineering
Washington University

www.arl.wustl.edu/~jst

Big Picture

- Minimum spanning trees
  - » general greedy method, Prim's algorithm, *d*-heaps
- Fibonacci heaps
  - » how operations are implemented, amortized analysis
- Shortest paths
  - » the shortest-path tree theorem and general labeling method
  - » topological scanning, Dijkstra's algorithm, breadth-first scanning
- Max flow problem
  - » general concepts and augmenting path method, including analysis
- Min cost flow problem
  - » shortest paths in graphs with negative edges
  - » negative cycles and cost reduction, least-cost augmenting path
  - » shortest paths using transformed edge costs
  - » min-cost augmenting paths using transformed edge costs
- Bipartite matching
  - » finding matchings using flows, augmenting paths in bipartite graphs

Min spanning trees and heaps

- State the greedy method
  - How is Prim's algorithm an implementation of the greedy method?
  - Given a connected graph with 100 vertices and 300 edges. Suppose we apply the red rule 201 times, what can we say about the uncolored edges?
- Given a graph and partial mst computed by Prim's algorithm
  - what vertices are in the heap?
  - for each vertex *u* in the heap, what is cheap(*u*), key(*u*)?
- In a 3-heap with 100 nodes, how many are 3 levels down from the root? How many siblings does "last node" have?

Washington University in St. Louis — Engineering

Fibonacci Heaps

- Can a F-heap with 1000 nodes have a node with rank 10? rank 15? Note: $\log_\phi n \approx 1.44 \lg n$
- For a specific F-heap, how many credits are needed to maintain the credit invariant?
- For an F-heaps data structure with 1000 nodes
  - » how many credits are needed right after initialization? after 500 melds? Give an upper bound on number of credits
- Consider a single F-heap on 1000 nodes
  - » Give a bound on the number of new trees created in the first step of a deletemin
  - » bound # of steps in the second part for which there is no collision
  - » bound # of credits needed to maintain invariant

4

Washington University in St. Louis Engineering

Shortest paths

- State the shortest path tree theorem
  - » suppose dist(u)=100, dist(v)=110 and length(u,v)=20 and the SPT edge joining v to its parent is removed from the graph; can you bound length of new shortest path to v?
- Why is scanning & labeling a variant of labeling method? How do we know it never misses an edge?
- Let $N_k(G,s)$ be the number of vertices in G that have a shortest path from s with $\leq k$ edges
  - » after 550 steps of the breadth-first scanning algorithm, how many vertices have $dist(u)$=shortest path distance if n=100?
- Given a digraph with positive edge weights, describe an algorithm to detect if it has a cycle for which the product of the edge weights is <1
  - » what is its running time?

5

Max flow problem

- What properties define a valid flow function?
  - » under what conditions is the sum of two flows a flow?
- Given a flow, what is its residual graph?
  - » identify a shortest augmenting path, how does the residual graph change when you add flow?
- In shortest augmenting path algorithm, if $level_i(t)=level_j(t)$, how many times can edge (u,v) be included in an augmenting path in steps $i+1$ to j
- Describe an algorithm to determine the edge connectivity of a graph (# of edges that can be deleted without disconnecting graph)
- Flow graphs with lower bounds on flows
  - » finding a feasible flow, turning feasible flow into max flow

Washington University in St.Louis | Engineering

Min-cost flows

- Given a graph and intermediate state for least-cost augmenting path algorithm
  - » show residual graph and least-cost spanning tree; find next path and update state; does new residual graph have negative cycles?
- In a graph with 200 vertices, 700 edges and a max-flow value of 137, a min-cost, max flow with value 52 and a max-cost max flow with value 300
  - » bound # of steps used by min-cost augmenting path algorithm
  - » bound # of neg. cycles found by cycle reduction method

Matchings

- Given a matching, show how to find a larger matching; or a matching of larger weight
- Determine if a given matching is max weight. Give a general method for determining if a matching in a bipartite graph has max weight
- Give an algorithm to find a max wt matching in a bipartite graph using the cycle reduction algorithm for min-cost flows
- Identify three augmenting paths with respect to a specific matching in a given graph
- For a given graph and matching, show a tree with root *x* that might be built by augmenting path algorithm; make tree as large as you can