CS 6200: Algorithmics II	**Fall 2020**
Instructor: Avah Banerjee, S&T.	
Lectures: 4	**Dates: 8/31**

1 Greedy Strategies and Matroids

Remark 1.1 *An algorithm is greedy if it solves an iterative problem, selecting the locally optimal solution at each stage with respect to the partial solution.*

Cormen et al.*defines the greedy algorithm as one that "makes a locally optimal choice in the hopes that this choice will lead to a globally optimal solution "[1]. The operative phrase is "in the hopes,"as greedy algorithms are not always optimal globally, and necessitate the following properties to be so:*
Optimal Substructure, *whereby the structure of the problem is such that separating it into distinct sub-problems would guarantee global improvement when one task is improved, and:*
Greedy Choice Property, *a generalization that choices made which are locally optimal are also globally optimal.*

1.1 Mimimum Spanning Forest Problem

Definition 1.2 *A forest is "an acyclic, undirected graph,"[1].*

Forests are generalized trees by this merit, without the "connected"stipulation, so the same algorithms that apply to trees also apply to forests as a function of repetition. In the Minimum Spanning Forest problem, a weight function w is applied to a graph $G(V(G), E(G))$ such that a total order is induced on the edges E(G). The goal is to find a spanning tree *for each component; that is, an acyclic graph in which all vertices are minimally connected, where each edge is a cut-edge. However, with the Minimum Spanning Forest problem, the sum total of weights $w(E)$ in the selected edges T used to form such a tree, is minimized.*

1.2 Borvka's Algorithm

Borvka's Algorithm is an iterative greedy algorithm which follows a pattern of reducing the search space with each step. First, the algorithm first selects each node's edge incident on it which has the least weight. Nodes in $V(G)$ connected by selected nodes are to be thought of as nodes in themselves, and the algorithm is repeated until only isolated vertices remain, so no more edges can be selected. A more formal definition of the algorithm is shown below:

Algorithm 1 Borvka's Algorithm

Input: G, T.
Output: returns G' composed of $V(G)$ and T.
if $E(G) == \emptyset$ **then**
  break
else
  for each $v \in V(G)$:
  choose $u \in V(G)$ such that the edge weight $w(u, v)$ is the minimum $\forall$ $E(G)$ which $(v, *)$
  $T = T \cup \{(u, v)\}$
  endloop
  $V(G')$ = components in $G(V(G), T)$
  $E(G')$ is a subset of $E(G)$ such that $E(G) \cup T = \emptyset$, and no edges between nodes in components defined by $G(V(G), T)$ are present.
  Borvka's Algorithm(G', T)
end if
return T.

1.3 Running Time of Borvka's Algorithm

During each iteration of the algorithm, each edge m is individually analyzed. Further, since one edge must be selected per vertex, and the worst possible case would be that the edges selected are shared between two vertices as connected components of size 2 with respect to T, the algorithm can iterate $log_2(n)$ times, where n is the number of vertices. This gives the algorithm a running time of $O(mlog_2n)$.

1.4 Correctness of Borvka's Algorithm

Lemma 1.1 *Optimal Substructure Let F be a minimum spanning forest of graph G. Let $A \subseteq F$ and $B = F - A$, where A is a subset of F, and B is a set of edges $E(G)$. Then, B is a minimum spanning forest of $G/A = G'$.*

Proof: *Suppose B does not constitute a minimum spanning forest of G'. Assume, then, that $\exists$ some $B*$ such that $w(B^*) < w(B)$. It follows that $w(B^* \cup A) < w(B \cup A)$. Note, $B^* \cup A$ is a spanning forest of G. But, $B \cup A = F$, a minimum spanning forest of G. By contradiction, no such B^* may exist.* ■

Lemma 1.2 *Greedy Choice Property For every $v \in V(G)$, the minimum weight edge incident to v is a part of some minimum spanning forest of G.*

Proof: ***CASE I:*** *Suppose that* $\exists$ *a minimum weight edge* e *joining two connected trees in a graph* G*, and that* $\exists$ *some minimum spanning forest* F *where* $e \notin$ *that minimum spanning forest* F*.*
CASE II: *Suppose that* e *and* e'*, an alternative to* e *that also joins two connected trees in* G*, and that* e' *and* e *are both incident on an arbitrary vertex* $v \in V(G)$*. Should* e *be selected to form a spanning forest* F*, and* e' *be selected to form a spanning forest* F'*,* $w(F) \leq w(F')$*. So,* e *would necessarily be selected in forming a minimum spanning tree.* ■

References and Further Reading

[1] *Cormen, T.,* et al.*(2009). Introduction to Algorithms. 3rd ed. Cambridge, Mass. MIT Press.*