Lecture Outline 2
Topics in Graph Algorithms (CSCI5320-22S)

CAI Leizhen
Department of Computer Science and Engineering
The Chinese University of Hong Kong
lcai@cse.cuhk.edu.hk

January 21, 2022

Minimum Spanning Tree

Keywords: Minimum spanning tree, algorithm **Greedy-MST**, cuts and cycles, blue/red rules, Kruskal's algorithm, Prim's algorithm, and Boruvka's algorithm.

Task: Find a spanning tree of minimum weight in a weighted connected graph $G = (V, E; w)$ with $w : E \to R$.

For a nonempty proper subset V' of vertices, $[V', V - V']$ denotes edges with one end in V' and the other end in $V - V'$, and is called a *cut*.

To obtain a spanning tree, we need to delete at least one edge in a cycle and keep at least one edge in a cut, which leads to the following nondeterministic paradigm of a greedy algorithm:

1. **Algorithm Greedy-MST**

   Repeatedly color each edge either blue (accept) or red (reject) according to the following two rules:

   (a) *Blue rule*: Choose a cut without blue edges and assign blue to an uncolored edge of minimum weight.

   (b) *Red rule*: Choose a cycle without red edges and assign red to an uncolored edge of maximum weight.

   When all edges are colored, blue edges form an MST.

2. *Correctness of* **Greedy-MST**

   To establish the correctness of **Greedy-MST**, we show that (1) all edges will be colored, and (2) (*color invariant*) there is an MST that contains all blue edges but no red edges.

   To prove (1), we show that either blue blue or red rule is applicable whenever there is an uncolored edge e.

Note that blue edges form blue components. If both ends of e lie in one blue component, then there is a cycle without red edges and the red rule is applicable to color e red. Otherwise the ends of e lie in two different blue components, and the blue rule is applicable to color an edge blue.

To prove (2), we can use induction on the number of colored edges. We assume that (2) is true right before an edge is colored, and prove that (2) remains true after an application of either blue or red rules.

Let T be an MST that satisfies the color invariant before an edge e is colored. We consider two cases depending on whether e is colored blue or red.

Case 1. *e is colored blue.* If e is contained in T, then we are done. Otherwise, e is not in T and we will construct a new MST T' that satisfies the color invariant. Consider the cut $[V', V - V']$ to which the blue rule is used to color e. In tree T there is a path joining the two ends of e, and hence there is an edge e' in T that is in $[V', V - V']$. Note that e' is neither blue (the blue rule is used for $[V', V - V']$) nor red (no edge in T is red). Therefore e' is uncolored and thus $w(e) \leq w(e')$, and $T' = T - e' + e$ is an MST satisfying the color invariant.

Case 2. *e is colored red.* Similar to Case 1 and is left as homework.

3. The following three classical algorithms are special cases of **Greedy-MST**.

MST-algorithm 1. *Kruskal's algorithm (1956)*

Apply the following step to edges in non-decreasing order by weight:

If the two ends of the current edge lie in two different blue trees, color it blue; otherwise color it red.

MST-algorithm 2. *Prim's algorithm (1957, Jarník 1930)*

Choose a vertex s as the starting vertex and repeat the following $n - 1$ times:

Let T be the blue tree containing s.

Select a minimum-weight edge in the cut $[V(T), V - V(T)]$ and color it blue.

MST-algorithm 3. *Boruvka's algorithm (1926)*

Repeat the following until there is a single blue tree:

For every blue tree T, select a minimum-weight edge in the cut $[V(T), V - V(T)]$.

Color all selected edges blue.

Note: All edge weights need to be distinct in order for Boruvka's algorithm to work correctly, otherwise we need a tie-breaking rule for edges of equal weight.

Question: Can we make changes to edge weights to make Boruvka's algorithm work correctly for general weighted graphs?

4. We can use **Greedy-MST** to generate more algorithms for MST.

   MST-algorithm 4. *Transforming from a spanning tree*

   Construct a spanning tree T of G, and for each edge e of G not in T apply the red rule to the unique cycle of $T + e$, and change T to the new spanning tree by removing the red edge from $T + e$.

   MST-algorithm 5. *Keeping cut edges*

   Consider edges in non-increasing order w.r.t. their weights. If the current edge e is not a cut edge for the current graph, then delete it.

   Question: More algorithms from **Greedy-MST**?

5. **References**

   [1] Tarjan, Data Structures and Network Algorithms, Chapter 6, SIAM, 1983