Lecture 20, Nov 22, 2023

Kruskal's MST Algorithm

- Kruskal's algorithm builds up the MST edge-by-edge
- Sort all edges in the forest in ascending order, and keep a spanning forest (starting from the trivial forest); for every edge in order of cost, if it connects two disjoint forests, include it in the MST; stop when we have exactly $n-1$ edges
    - The algorithm can be proven correct by induction, making use of the MST construction theorem
- Use a union-find structure to keep track of the partition of nodes into minimum spanning forests
- Often times we don't need to go through all edges; to optimize the algorithm we can use a heap and extract only the edges as necessary
    - The initial call to HEAPIFY() takes only $O(m)$, and each subsequent extraction takes around $O(\log m)$
    - In the worst case where we need to look at every edge, this still takes $O(m \log m)$, but if we only need a fraction of edges this can be substantially faster
- Formally: given a connected, undirected, weighted graph $G = (V, E)$ where $V = \{ 1, 2, \dots, n \}$ and E is an array of m weighted edges:
    1. Turn E into a min-heap
    2. For $i = 1$ to n, MAKE-SET(i) (initialize disjoint sets)
    3. Initialize the set of MST edges with an empty set
    4. While we have less than $n-1$ edges in the MST, do:
        1. Extract the edge (u, v) with minimum weight from E
        2. Find the representatives of the sets containing u, v; if they are different:
            1. Merge the two sets
            2. Add the edge (u, v) to the MST
- Complexity analysis:
    - Building the min-heap takes $O(m)$
    - Making the n sets takes $O(n)$
    - In the worst case, the loop runs m times since we have to go through all edges
    - All the heap extractions then take $O(m \log n)$ (note $O(m \log n) = O(m \log m)$, since $m \leq n^2$)
    - We also have $n-1$ unions and at most $2m$ finds, giving a complexity of $O(m \log^* n)$ (using a disjoint-set forest with path compression and union by size)
    - Therefore the total complexity is $O(m \log n)$