CMPSC112
Lecture 32: Shortest Paths

Prof. John Wenskovitch

04/18/2016
Last Time

- Edge-Weighted Graphs
  - Definitions and storage
- Spanning Trees
  - Prim’s Algorithm
  - Kruskal’s Algorithm
But Wait, We Did Shortest Path with BFS...

- Yes, we did, but now our edges have weights!
  - A path that takes 4 edges could have a smaller total weight than a path that takes 2 edges.

- A **shortest path** from vertex $s$ to vertex $t$ in an edge-weighted digraph is a directed path from $s$ to $t$ with the property that no other such path has a lower weight.
Shortest Path Assumptions

• Paths are directed. Undirected edges can be considered as a pair of directed edges. We must respect the direction shown by the edge.

• Weights are (again) not necessarily distances, nor are they necessarily costs (but they could be).

• Not all vertices need to be reachable. If no path exists from $s$ to $t$, then it follows that there is no shortest path.

• Negative weights make things tough, so we’ll assume all weights are positive or zero.

• Shortest paths are not necessarily unique.
Dijkstra’s Algorithm

• Basic idea:
  – Initialize \texttt{distTo}[s] (starting vertex) to 0, and \texttt{distTo[everything_else]} to positive infinity.
  – Add to the tree a non-tree vertex with the lowest \texttt{distTo[\cdot]} value.
  – Repeat until all vertices are on the tree, or no non-tree vertex has a finite \texttt{distTo[\cdot]} value.
Dijkstra’s Algorithm – Visual
Any Questions?