Lecture 14 – Greedy Algorithms in Graphs
Reading: KT Sections 4.4 and 4.5

Partial content of these slides have been obtained from the official lecture slides that accompany the textbook. A complete set of slides can be found at: http://www.cs.princeton.edu/~wayne/kleinberg-tardos/
Shortest path problem

- Input:
  - Directed graph $G=(V,E)$
  - Weight function $w$

- What’s the weight of a path?
- What’s a shortest path?

- What’s the output of that problem?

Example*

\[
\begin{array}{cccccccc}
0 & 6 & \infty & \infty & \infty & \infty & \infty & \infty \\
\infty & 2 & 3 & 2 & \infty & \infty & \infty & \infty \\
\infty & \infty & 7 & \infty & \infty & \infty & \infty & \infty \\
\infty & \infty & \infty & 2 & 4 & 9 & \infty & \infty \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
\infty & \infty & \infty & \infty & \infty & \infty & \infty & \infty \\
\end{array}
\]
Example*

---

Example*
Example*

What would happen if we have negative edges?
Dijkstra’s “water-ripple” idea

• Start with initial vertex in visited set
• Maintain sets of visited vs. unvisited vertices
• Expand visited set one vertex at a time
• Look at all arcs from visited set to any vertex in unvisited set.
• Choose the vertex that is cheapest to include and visit it

Dijkstra’s algorithm

Dijkstra’s Algorithm \((G, \ell)\)
Let \(S\) be the set of explored nodes
  For each \(u \in S\), we store a distance \(d(u)\)
Initially \(S = \{s\}\) and \(d(s) = 0\)
While \(S \neq V\)
  Select a node \(v \not\in S\) with at least one edge from \(S\) for which
  \(d'(v) = \min_{e=(u,v), u \in S} d(u) + \ell_e\) is as small as possible
  Add \(v\) to \(S\) and define \(d(v) = d'(v)\)
EndWhile

Why does this work?