Week 11: Shortest Path Algorithms

Day 18 (M 3/30): Intro to Graphs

- **Video (26 min)**: Watch the following video introducing Dijkstra's algorithm. `https://youtu.be/dzroIeaS_ek`
- **Exercise (10 min)**: This exercise will give a proof of correctness for Dijkstra's algorithm. After running the algorithm, suppose that we define an adjusted weight for every edge (u, v) as

$$w'(u, v) = w(u, v) + \text{dist}[u] - \text{dist}[v],$$

where $dist[u]$ is the labeled distance at the end of Dijkstra's algorithm.

For any path from i to j, we can show that the adjusted distance of the path is equal to the unadjusted distance $+p(i) - p(j)$. Since the latter is a constant for any path from i to j, this shows that the shortest path from i to j with the adjusted weights is the same as for the original weights. (Try this if you have time.)

1. Show that $w'(u, v) \geq 0$ for all edges and that $w'(u, v) = 0$ if $\text{prev}[u] = \text{prev}[v]$.

2. Explain why combining the above two results, proves correctness of the algorithm.

- **Video (13 min):** Watch the following video talking about correctness and runtime of Dijkstra's algorithm. `https://youtu.be/s04tMSFaVZY`
- **Exercise (10 min):** Explain how to use a min-heap to achieve $O(|E| \log(|V|))$ runtime for Dijkstra's algorithm.

- **Post any questions about this section to the corresponding Canvas discussion thread prior to the live discussion.**

Day 19 (R 4/2): Bellman-Ford and Dynamic Programming

- **Video (20 min)**: Watch the following video introducing Bellman-Ford's algorithm. `https://youtu.be/dYd5ZMDibrY`
- **Exercise (10 min)**: Give an algorithm to detect whether or not there is a negative cycle in a graph. **Hint:** What would happen in Bellman-Ford if there was?

- **Video (15 min):** Watch the following video about dynamic programming. `https://youtu.be/i_Ez_aZbRRc`
- **Exercise (10 min)**: This question relates to the dynamic programming solution to the knapsack problem given in the video above.

  1. Give the runtime of calculating all $V_i[w]$ values. You may assume W and all w_i values are non-negative integers.

  2. Explain how you would use the calculation of the $V_i[w]$ values to find the optimal solution to the knapsack problem.

- **Post any questions about this section to the corresponding Canvas discussion thread prior to the live discussion.**

Extra Resources for the Week

- Dijkstra's Algorithm: `https://medium.com/basecs/finding-the-shortest-path-with-a-little-help-from-dijkstra-`
- Bellman-Ford Algorithm: `https://web.stanford.edu/class/archive/cs/cs161/cs161.1182/Lectures/Lecture12/Lecture12-compressed.pdf`
- Dijkstra's Visualization: `https://www-m9.ma.tum.de/graph-algorithms/spp-dijkstra/index_en.html`
- Bellman-Ford Visualization: `https://www-m9.ma.tum.de/graph-algorithms/spp-bellman-ford/index_en.html`