Problem Set 4

CS 331H

Due Monday, March 4

1. Work through the Jupyter notebook on the class website.

2. Consider a weighted, directed graph where all distances lie in [1, 2). We would like to find an $O(m)$ time algorithm for single-source shortest paths on this graph, where $m$ is the number of edges.

   (a) Consider a variant of Dijkstra’s algorithm that does not always visit the unvisited node of smallest $c(u)$, but instead arbitrarily picks one of the unvisited nodes of smallest $\lfloor c(u) \rfloor$. Show that such an algorithm still yields the correct answer.

   (b) Now give a data structure that allows this Dijkstra variant to run in $O(m)$ time. **Hint:** at any point during the execution, the set of $\lfloor c(u) \rfloor$ for unvisited nodes $u$ can only have a small number of options.

   (c) Extend your result to $O(mC)$ time and $O(m)$ space for distances in $[1, C)$ for any $C \geq 1$. 
