Algorithm: Karp-Edmonds

For the Knight's Max Flow project, we attempted to utilize the Karp-Edmonds algorithm. The objective of this algorithm is to use Breadth First Search to discover augmenting paths and update the flow values after each one is found. Thus, we acquired a source and a sink value and built a BFS tree. Our algorithm would not discover any augmenting paths if our search does not reach the sink from the first connected component. However, if our algorithm discovers any augmenting path, it will update the edges.

Randomization played a key role in our algorithm's edge placement. From the source, we performed BFS to pave a path to the sink. As we start from the source, we search for a list of its neighbors. Once we have that list, we randomly select a neighbor and add it to a queue of visited cells. This queue essentially represents our arbitrary path to the sink. After choosing a random neighbor, we take the average of the two cell's capacities, calculate the edge capacity, then draw the edge to help solidify our path to the sink.

To calculate graph flow, we essentially searched through our list of touched edges to find the edge with the smallest capacity, also known as the bottleneck value. Consequently, we augmented the rest of the path by adding bottleneck value to the rest of the edge capacities.