CS 3510

Ryan Soedjak **TSP Project** 4/19/19

The TSP algorithm implemented is the simulated annealing

```
annealing(graph $G$, time limit $t$):
    Input: a complete graph G
    Input: the time limit t
    Output: A cycle G
    sortedEdgeList ← edges of $G$ in sorted order by weight
    for edge $e$ in sortedEdgeList in decreasing order:
        if hasCycle($G$, $e$):
            search through the adjacency list of $G$ and remove $e$
    return $G$

hasCycle(graph $G$, edge $e$):
    Input: an undirected connected graph $G = (V, E)$, edge $e = (v_1, v_2) \in E$
    Output: true if G has a cycle with edge e, false otherwise
    This runs DFS on $G' = (V, E - \{e\})$ starting at $v_1$ and sees if it ever reaches $v_2$
```