MCS 253P - Lab 6

You will be implementing a function that can calculates the single source preferred path to a specific goal using ***graph search (ie. we keep track of expanded nodes, and don’t re-expand them unless a cheaper path is found),***  that utilizes either Dijkstra's algorithm, greedy heuristic search, or A\*, as requested.

Your function should take the following as arguments:

* information about the starting node
* information about the goal node
* a graph
* a string indicating the type of search. Accepted strings are:
  + “Greedy”
  + “Dijkstra”
  + “A\*”

Your function should return a ***pair*** object as follows:

* pair.first is a string storing the calculated path from the starting node to the ending node (inclusive)
* pair.second is a string storing each node that was expanded (ie. extracted from the queue) during the search in order

If you implement your function correctly, there should really be only one difference between the different search methods!!!

You main will load a graph, ask for the type of search from the user, then print out the calculated path and the expanded nodes (in order) for that search.

Input:

* graph (will be provided via text file, but you can choose whether to load it via stdin or load it from a file)
* start node
* end node
* type of search (“Greedy”, “Dijkstra”, “A\*”)

Output:

* for each search…
  + calculated path
  + expanded nodes

Example:

input format:

[number of nodes]

[name of node 1] [heuristic value for node 1]

[name of node 2] [heuristic value for node 2]

.

.

.

[number of edges]

[connecting node 1] [connecting node 2] [edge weight]

[connecting node 1] [connecting node 2] [edge weight]

.

.

.

input graph:

[lab6\_sampleGraph.txt](https://drive.google.com/open?id=1twIOCilyT02vxWxbg3R0Q8rPrSBLtx3I)

user input:

Start

Goal

Greedy

Start

Goal

Dijkstra

Start

Goal

A\*

output:

*(figured out by hand in lab!)*