**Unit 3 Algorithmics**

**Submit Task – Week 6**

Prim’s Algorithm

Using the classes provided, construct Prim’s algorithm in Python. Include comments to serve as pseudocode. Note the set\_colour function has been included to allow you to set the colour to show your tree. Alternatively you can print a list of edges.

Template: <https://trinket.io/python3/70e28e0841>

Submit your PY file and a screenshot of your output.

Further Questions

1. How can a priority queue be used in Prim’s algorithm to improve efficiency?
   1. By using a P queue, you can efficiently find the minimum weighted edge among all the edges that connect the vertices in the MST to the vertices outside of the tree.
      1. Start with an empty priority queue, add all the vertices to the queue, with a key (infinity) except for the starting vertex which has a priority of 0.
      2. Repeat until all vertices are included in the MST
         1. Find all the vertices with the minimum priority from the queue
         2. Mark the extracted vertex as visited
         3. Update the priorities of extracted vertices (If vertex is not visit, and weight of edge connecting it top extracted vertex xis smaller than its current priority update its priority to the weight of the edge.
      3. When all vertices are visited the MST is constructed
2. Assuming all edge weights are positive, under what conditions will an edge with the smallest weight not appear in the resulting tree?
   1. When it forms a cycle?
3. Explain why Prim’s algorithm is ‘greedy’.
   1. Its cause it selects the most locally optimal step at any given stage with the hope of finding a globally optimum solution (MST).
4. Will Prim’s work with negative edges? Explain why/why not.
   1. It won’t as prims relies on selecting the minimum weight of edges at each step which may not hold true when negative edge weights are involved. It can also lead to the formation of cycles that decrease the overall weight