

One-Day Assignment 9 – algorithmic solution

Lost Map

One Day Assignment 9 – General

Run MST.

Additional notes:

For Prim's:

Consider storing an IntegerTriple in your PQ (store w, u, v), so you can keep track of the exact edges in the MST, since the problem requires the edges themselves, not just the total cost. Converting the graph to an Adjacency list not strictly necessary.

For Kruskal's:

Conversion to an Edge List would be needed, but as a default Edge List stores u and v already, getting the edges themselves may be easier than modifying Prim's.

One Day Assignment 9 – General

For Prim's variant for dense graphs:

Lecture slides example already uses the version which stores the edges themselves, which can be modified to output the edges instead

One Day Assignment 9 – Prim's

convert AdjMat to AdjList (optional)

set vertex 1 (or 0, depending on numbering) to taken; add neighbours to PQ of IntegerTriples

while PQ is not empty:

- dequeue edge from PQ

- if v of edge is not taken:

 - output edge (u, v)

 - set v to taken

 - add neighbours of v that are not taken to the PQ

One Day Assignment 9 – Prim's Variant

convert AdjMat to AdjList (optional)

initialise IntegerPair array A (set everything to (inf, -1)), and boolean array B (set everything to false (already done implicitly))

set A[1] (or A[0], depending on numbering) to (0, -1)

One Day Assignment 9 – Prim's Variant

while true:

 scan through A to find the pair with the smallest first value, as well as its index

 if pair.first is inf, break

 if pair.second is not -1:

 output (index, pair.second)

 B[index] = true; A[index] = inf

 for all outgoing edges (v, w) from index:

 if (!B[v] and A[v].first > w):

 A[v] = (w, index)

One Day Assignment 9 – Kruskal's

convert to edge list; sort edge list in ascending order of weight

set up UFDS

for edge in edge list (left to right):

 if (u, v) are not in the same set:

 union sets containing u and v

 output edge (u, v)