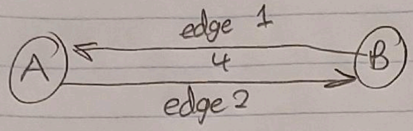
My graph class makes undirected graphs by connecting nodes with 2 edges.



edge 1) weight=4 source=B destination=A

edge 2) weight=4 source=A destination=B

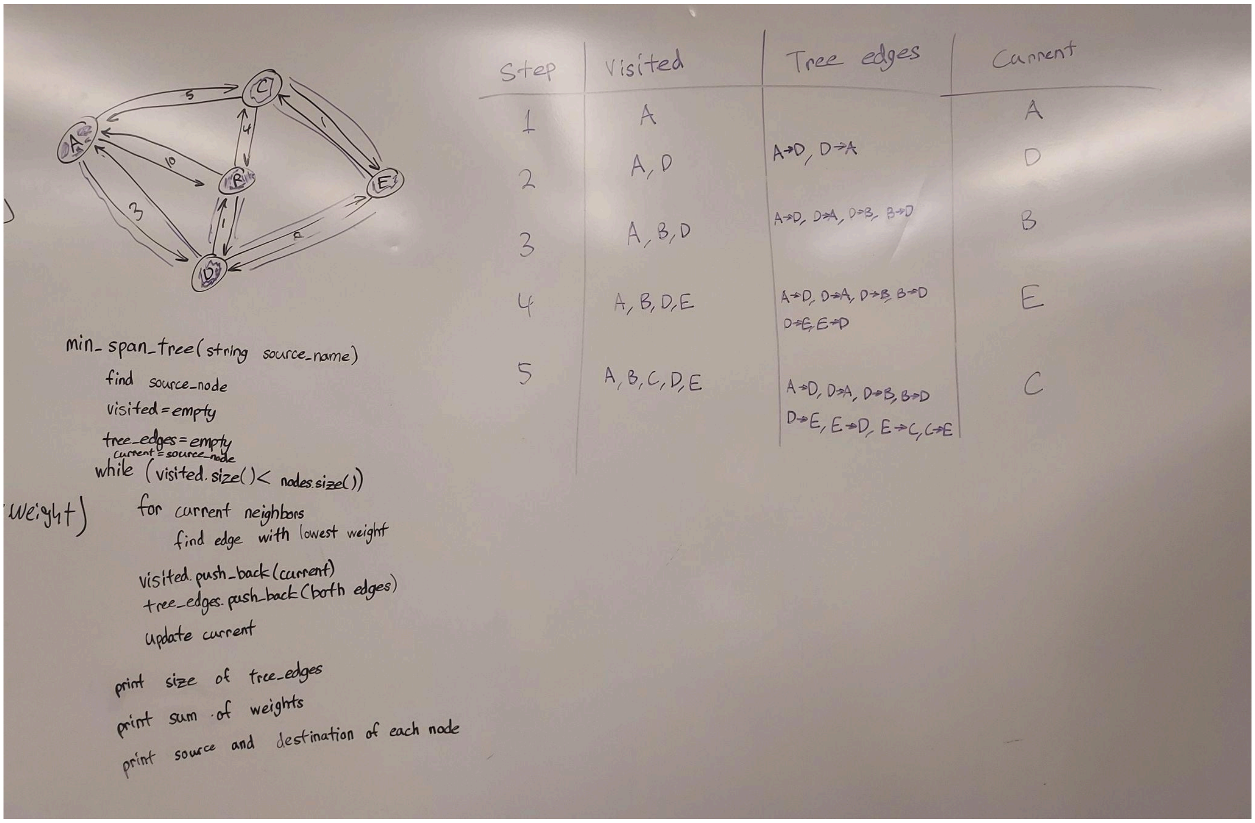
Prim's Algorithm

- Finds the lightest tree that touches all nodes.
- Work's for undirected graphs.

Desired Output

My function will output both edges, but will count them as 1 edge.

alphabet_graph_2

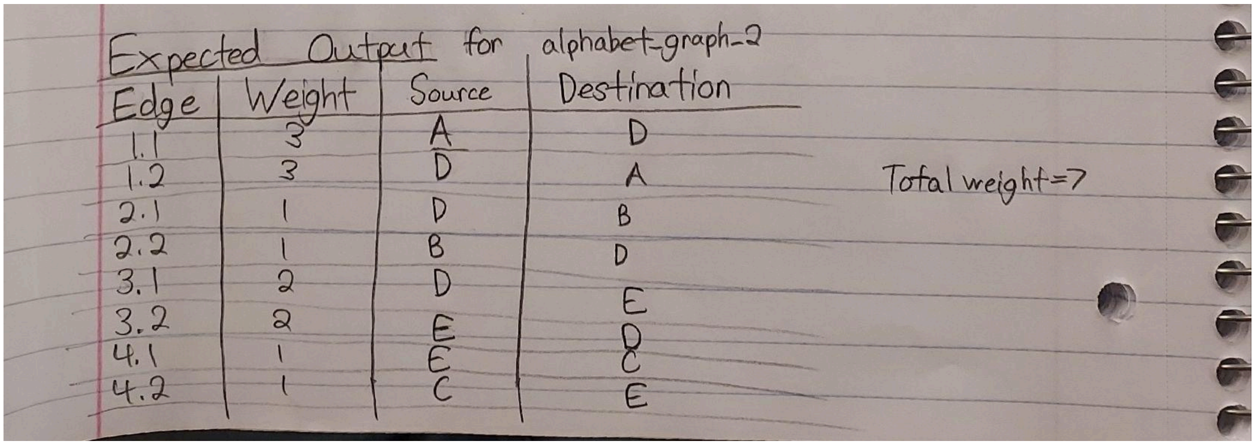


```
min_span_tree(string source_name)
    find source_node
    visited=empty
    tree_edges=empty
    current=source_node
  while (visited.size()< nodes.size())
      for current neighbors
          find edge with lowest weight

      visited.push_back(current)
      tree_edges.push_back(both edges)
      update current

    print size of tree_edges
    print sum of weights
    print source and destination of each node
```

Step	Visited	Tree edges	Current
1	A		A
2	A, D	A→D, D→A	D
3	A, B, D	A→D, D→A, D→B, B→D	B
4	A, B, D, E	A→D, D→A, D→B, B→D D→E, E→D	E
5	A, B, C, D, E	A→D, D→A, D→B, B→D D→E, E→D, E→C, C→E	C

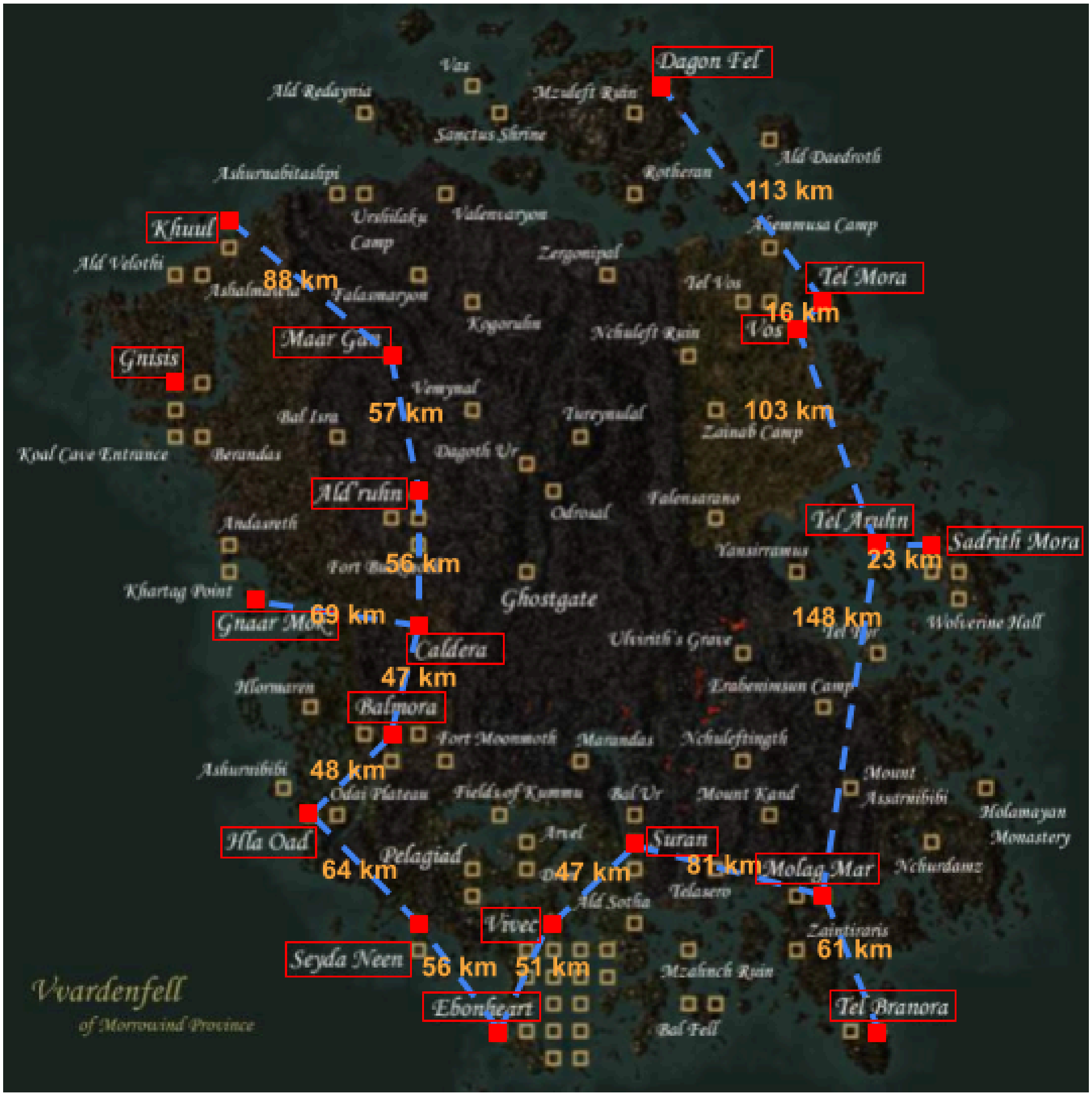


Expected Output for alphabet_graph_2

Edge	Weight	Source	Destination
1.1	3	A	D
1.2	3	D	A
2.1	1	D	B
2.2	1	B	D
3.1	2	D	E
3.2	2	E	D
4.1	1	E	C
4.2	1	C	E

Total weight=7

Expected minimum spanning tree for vvardenfell_graph



Total weight = 1128 km