**Generate\_greedy\_2regret\_solution(dist\_matrix, start\_node):**

*num\_nodes = dist\_matrix.shape[0]*

*num\_select = (num\_nodes + 1) // 2*

*selected\_nodes = [start\_node]*

*unselected\_nodes = {num\_nodes} \ {start\_node}*

*total\_distance = 0*

***WHILE*** *len(selected\_nodes) < num\_select:*

*regret\_node = None*

*regret\_position = None*

*best\_regret= float('-inf')*

*regret\_best\_increase = float('inf')*

***FOR*** *node in unselected\_nodes:*

*best\_node = None*

*best\_position = None*

*best\_min\_increase =infinity*

*second\_best\_min\_increase = infinitiy*

***FOR*** *i in range(len(selected\_nodes):*

*next\_index = (i +1)% len(selected\_nodes)*

*increase = (dist\_matrix[selected\_nodes[i], node] +*

*dist\_matrix[node, selected\_nodes[next\_i]] -*

*dist\_matrix[selected\_nodes[i], selected\_nodes[next\_i]])*

*// calculate the increase of distance if the node is inserted after the node of index i*

***IF*** *increase < second\_best\_min\_increase:*

***IF*** *increase < best\_min\_increase:*

*best\_min\_increase = increase*

*best\_node = node*

*best\_position = next\_i*

***ELSE****:*

*second\_best\_min\_increase = increase*

*regret= second\_best\_min\_increase - best\_min\_increase*

***IF*** *regret > best\_regret:*

*best\_regret = regret*

*regret\_node = best\_node*

*regret\_position = best\_position*

*regret\_best\_increase = best\_min\_increase*

***ADD*** *regret\_node at the regret\_position to selected nodes*

***REMOVE*** *regret\_node from unselected\_nodes total\_distance += regret\_best\_increase*

*total\_distance += dist\_matrix[selected\_nodes[-1], selected\_nodes[0]]*

***return*** *selected\_nodes, total\_distance*

**Generate\_greedy\_2regret\_weights\_solution(dist\_matrix, start\_node, weight):**

*num\_nodes = dist\_matrix.shape[0]*

*num\_select = (num\_nodes + 1) // 2*

*selected\_nodes = [start\_node]*

*unselected\_nodes = {num\_nodes} \ {start\_node}*

*total\_distance = 0*

***WHILE*** *len(selected\_nodes) < num\_select:*

*regret\_node = None*

*regret\_position = None*

*best\_regret= float('-inf')*

*regret\_best\_increase = float('inf')*

***FOR*** *node in unselected\_nodes:*

*best\_node = None*

*best\_position = None*

*best\_min\_increase =infinity*

*second\_best\_min\_increase = infinitiy*

***FOR*** *i in range(len(selected\_nodes):*

*next\_index = (i +1)% len(selected\_nodes)*

*increase = (dist\_matrix[selected\_nodes[i], node] +*

*dist\_matrix[node, selected\_nodes[next\_i]] -*

*dist\_matrix[selected\_nodes[i], selected\_nodes[next\_i]])*

*// calculate the increase of distance if the node is inserted after the node of index i*

***IF*** *increase < second\_best\_min\_increase:*

***IF*** *increase < best\_min\_increase:*

*best\_min\_increase = increase*

*best\_node = node*

*best\_position = next\_i*

***ELSE****:*

*second\_best\_min\_increase = increase*

*regret= second\_best\_min\_increase - best\_min\_increase*

*score = weight \* regret - (1-weight)\*best\_min\_increase*

***IF*** *score > best\_score:*

*best\_score = score*

*score\_node = best\_node*

*score\_position = best\_position*

*score\_best\_increase = best\_min\_increase*

***ADD*** *score\_node at the score\_position to selected nodes*

***REMOVE*** *score\_node from unselected\_nodes total\_distance += score\_best\_increase*

*total\_distance += dist\_matrix[selected\_nodes[-1], selected\_nodes[0]]*

***return*** *selected\_nodes, total\_distance*