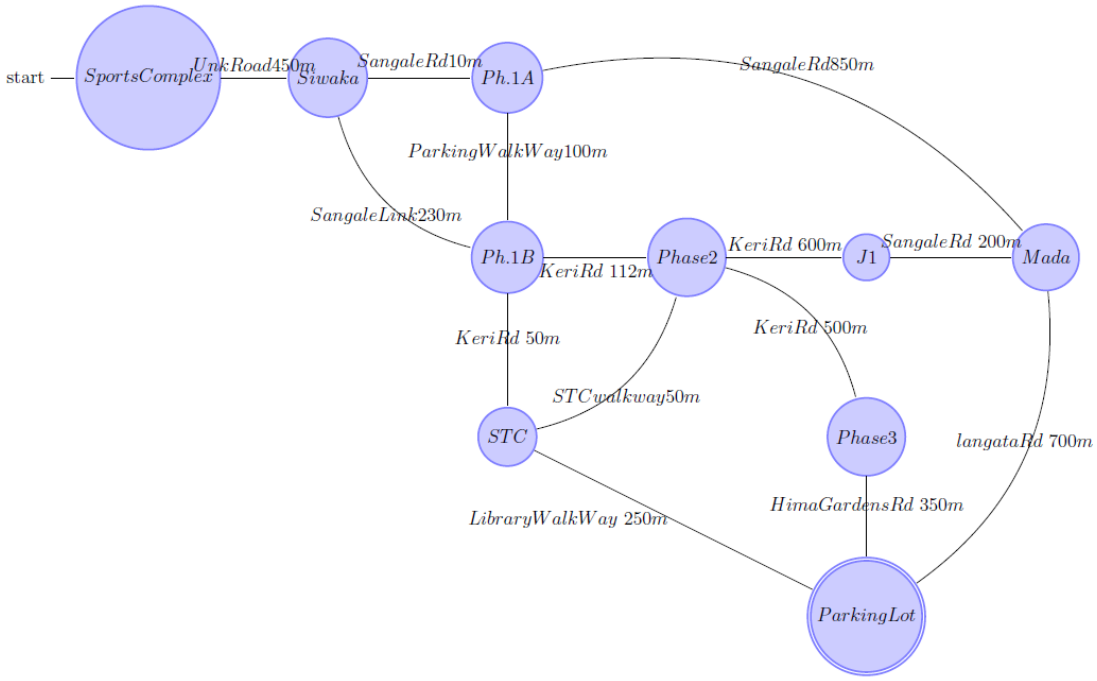
AI Task One: Python Implementation of Search Strategies for Route Finding Problem

Instructions: ***Work in Groups of max 5***

Upload your solutions on group leaders GitHub deadline Today Monday 12th 2020 midnight

1. Using **networkx** and **matplotlib** python libraries construct a graphical representation of the below Madaraka Estate Network **(5 marks)**



2. Implement a **Greedy BFS (Best First Search)** algorithms class pseudocode below **(Already Done in Class)**

```
function BREADTH-FIRST-SEARCH(problem) returns a solution, or failure
  node ← a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
  if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
  frontier ← a FIFO queue with node as the only element
  explored ← an empty set
  loop do
    if EMPTY?(frontier) then return failure
    node ← POP(frontier)  /* chooses the shallowest node in frontier */
    add node.STATE to explored
    for each action in problem.ACTIONS(node.STATE) do
      child ← CHILD-NODE(problem, node, action)
      if child.STATE is not in explored or frontier then
        if problem.GOAL-TEST(child.STATE) then return SOLUTION(child)
        frontier ← INSERT(child, frontier)
```

3. Call the Greedy BFS function and **color** all the **nodes** that the algorithm has visited **(5 marks)**

4. Implement class **UCS** pseudocode below and use it to walk a robot in the below graph from *Strathmore Sports Complex* to the *Parking Lot* **color** the nodes. **(10 marks)**

```
function UNIFORM-COST-SEARCH(problem) returns a solution, or failure
  node ← a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
  frontier ← a priority queue ordered by PATH-COST, with node as the only element
  explored ← an empty set
  loop do
    if EMPTY?(frontier) then return failure
    node ← POP(frontier)  /* chooses the lowest-cost node in frontier */
    if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
    add node.STATE to explored
    for each action in problem.ACTIONS(node.STATE) do
      child ← CHILD-NODE(problem, node, action)
      if child.STATE is not in explored or frontier then
        frontier ← INSERT(child, frontier)
      else if child.STATE is in frontier with higher PATH-COST then
        replace that frontier node with child
```

Vertex	h_{SLD}
Strathmore Sports Complex	730m
Siwaka	405m
Phase 1 Entrance A	380m
Phase 1 Entrance B	280m
STC	213m
Phase 2	210m
J1	500m
Phase 3	160m
Mada	630m
Parking Lot	0m