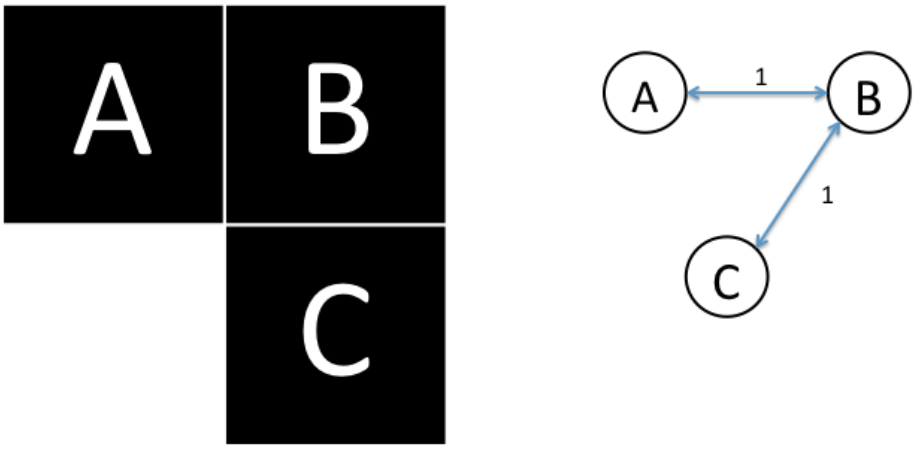
The National University of Singapore
School of Computing
CS3247 – Game Development
Homework Quiz 1 (Individual)
(Topics: Path-finding and Decision Making)

Submit solutions to Canvas->Homwwork Quiz1 by Thursday, 22 Feb 2024 [11.59 PM].

- **It can be typed or handwritten and scanned copy.**
- **You will be asked to explain your solutions to the class during the lecture session which comes after the due date.**

1 The following figure shows a tile map and its graph representation to run the path-finding algorithm. The cost to move to an adjacent cell is 1 unit. Only horizontal and vertical movements are allowed. One of the Non-Player Characters (NPC) in the game is huge and it cannot turn quickly. Hence, the developers decided to create additional tiles to represent orientations in path finding. In each tile, the NPC can have four different orientations: up, down, right and left. The cost to turn 90^0 angles is 1 unit.



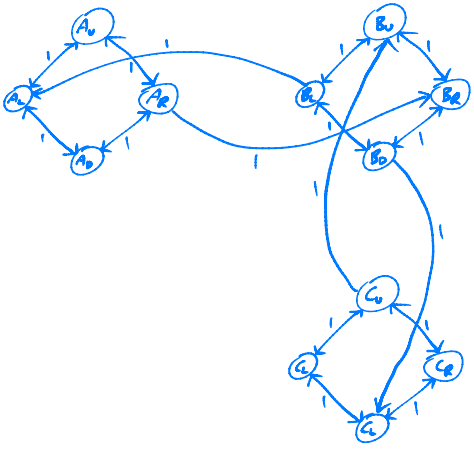
Game map and its graph representation

a) Draw the graph showing 12 nodes representing the positions and orientations for this large size NPC.

b) Assuming 'A' as the position (tile) of the large size NPC and 'up' (north) as its orientation, find the total cost using A* algorithm for moving from A to C, facing 'up' (north) direction in C. Use Manhattan distance (without considering orientation) between tiles as heuristic. To keep it simple, assume that ONLY clock-wise turning. Show the open-list after each iteration in a table form. [In the following table, 13: $S_2 \leftarrow S_1$ means reaching node S2 through S1 at the estimated total cost of 13].

Step	Open list
1	10: S_1
2	13: $S_2 \leftarrow S_1$, 15: $S_3 \leftarrow S_1$
…	…

1. a)



b)

Step	Open list
1	$2: A_U$
2	$3: A_R \leftarrow A_U$
3	$3: B_R \leftarrow A_R \leftarrow A_U$, $4: A_D \leftarrow A_R \leftarrow A_U$
4	$4: B_D \leftarrow B_R \leftarrow A_R \leftarrow A_U$, $4: A_D \leftarrow A_R \leftarrow A_U$
5	$4: C_D \leftarrow B_D \leftarrow B_R \leftarrow A_R \leftarrow A_U$, $4: A_D \leftarrow A_R \leftarrow A_U$, $5: B_L \leftarrow B_D \leftarrow B_R \leftarrow A_R \leftarrow A_U$
6	$5: C_L \leftarrow C_D \leftarrow B_D \leftarrow B_R \leftarrow A_R \leftarrow A_U$, $4: A_D \leftarrow A_R \leftarrow A_U$, $5: B_L \leftarrow B_D \leftarrow B_R \leftarrow A_R \leftarrow A_U$
7	$6: C_U \leftarrow C_L \leftarrow C_D \leftarrow B_D \leftarrow B_R \leftarrow A_R \leftarrow A_U$, $4: A_D \leftarrow A_R \leftarrow A_U$, $5: B_L \leftarrow B_D \leftarrow B_R \leftarrow A_R \leftarrow A_U$

Total cost = 6

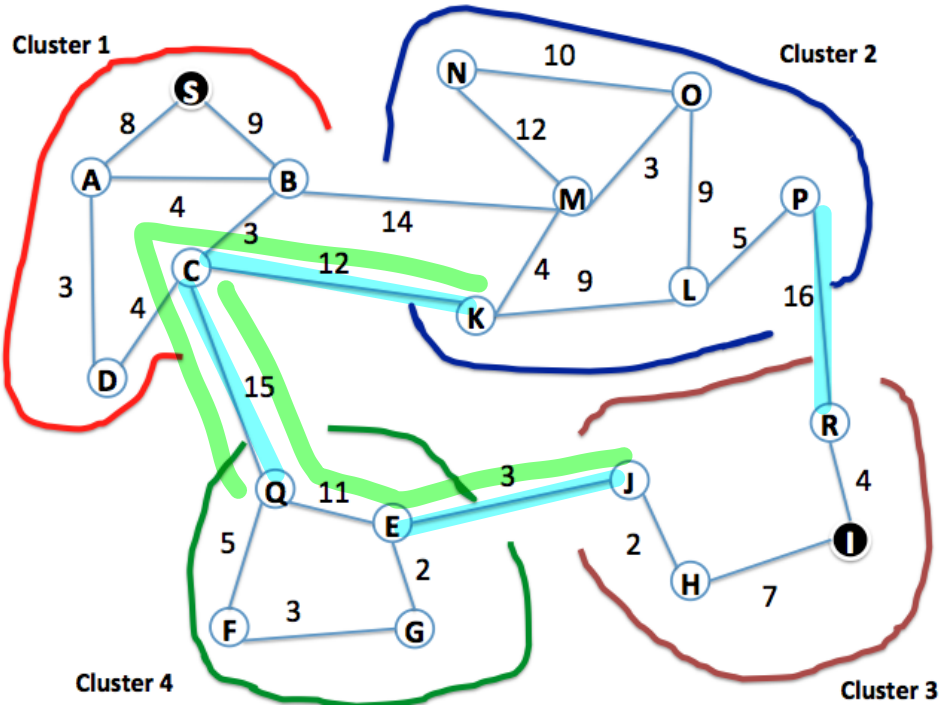
c) i) $h(n)$ = movement cost
$h^*(n)$ = movement cost + turning cost, turning cost ≥ 0
$h(n) \leq h^*(n)$

Hence, the heuristic is admissible.

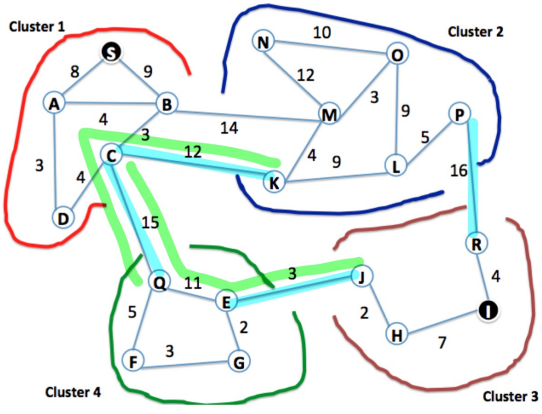
ii) Let n, n' be any 2 nodes such that there exists a path from n to n'
$c(n, n') = 1$ since all possible moves i.e. moving and rotating costs 1
$h(n) - h(n') \leq 1$ since:
1. The player can only move 1 tile at a time horizontally or vertically
2. The heuristic measures total vertical + horizontal dist.

$h(n) - h(n') \leq c(n, n')$
$h(n) \leq c(n, n') + h(n')$
Hence, the heuristic is consistent.

c)	i) Is the Manhattan distance (stated above in 'b') admissible heuristic? ii) Is the Manhattan distance (stated above in 'b') monotonic heuristic with respect to the 12 nodes graph? (refer slides: 74, 75)

2.	A game map is represented as nodes and clusters as shown below. A bot standing at node S (start node) needs to find its way to the node I (destination node). A* algorithm with cluster heuristic is used to find the path. The cost of the shortest path from a node in the current cluster (initially this will be the source cluster) to a node in destination cluster as the heuristics between the nodes in these two clusters. 
a)	Find the heuristic between the clusters and fill-in a table as shown below. Note: This should be computed manually. In reality, you should run A* several rounds (offline) to find the cluster heuristics.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	0	12	29	15
Cluster 2	12	0	16	27
Cluster 3	29	16	0	3
Cluster 4	15	27	3	0



	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Cluster 1	0	12	29	15
Cluster 2	12	0	16	27
Cluster 3	29	16	0	3
Cluster 4	15	27	3	0

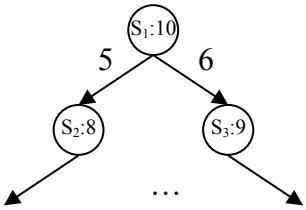
b)

Step	Open list
1	29: S
2	37: A←S, 38: B←S
3	38: B←S, 40: D←A←S
4	39: M←B←S, 40: D←A←S, 41: C←B←S
5	40: D←A←S, 41: C←B←S, 42: O←M←B←S, 43: K←M←B←S, 51: N←M←B←S
6	41: C←B←S, 42: O←M←B←S, 43: K←M←B←S, 51: N←M←B←S
7	30: Q←C←B←S, 40: K←C←B←S, 42: O←M←B←S, 51: N←M←B←S
8	35: F←Q←C←B←S, 40: K←C←B←S, 41: E←Q←C←B←S, 42: O←M←B←S, 51: N←M←B←S
9	38: G←F←Q←C←B←S, 40: K←C←B←S, 41: E←Q←C←B←S, 42: O←M←B←S, 51: N←M←B←S
10	40: K←C←B←S, 40: E←G←F←Q←C←B←S, 42: O←M←B←S, 51: N←M←B←S
11	40: E←G←F←Q←C←B←S, 42: O←M←B←S, 49: L←K←C←B←S, 51: N←M←B←S
12	40: J←E←G←F←Q←C←B←S, 42: O←M←B←S, 49: L←K←C←B←S, 51: N←M←B←S
13	42: O←M←B←S, 42: H←J←E←G←F←Q←C←B←S, 49: L←K←C←B←S, 51: N←M←B←S
14	42: H←J←E←G←F←Q←C←B←S, 49: L←K←C←B←S, 51: N←M←B←S
15	49: L←K←C←B←S, 49: I←H←J←E←G←F←Q←C←B←S, 51: N←M←B←S
16	49: I←H←J←E←G←F←Q←C←B←S, 51: N←M←B←S, 54: P←L←K←C←B←S

min. path : S→B→C→Q→F→G→E→J→H→I

cost : 49

b) Use A* algorithm to find the path with the minimum cost for the character to move from the node S to the node I. Provide a table list of open nodes at each step similar to the table provided for the graph shown below.



Step	Open list
1	10: S_1
2	13: $S_2 \leftarrow S_1$, 15: $S_3 \leftarrow S_1$
…	…

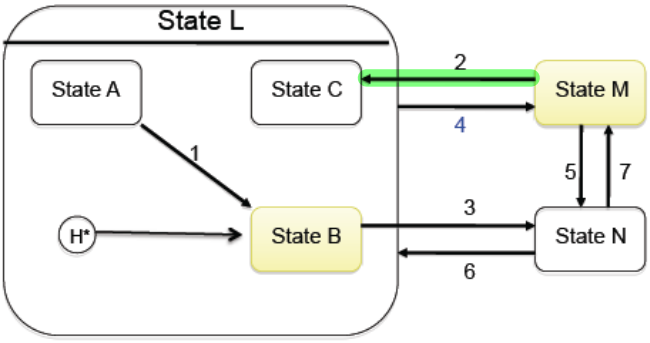
3.

State Machine: What will be the Target State and Action sequence if Transition 2 is triggered after Example 3 in Lecture slides (slide 36)?

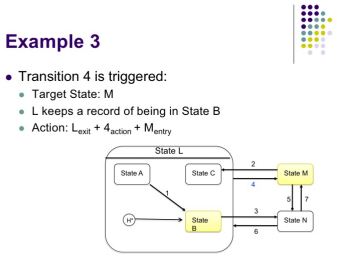
Target State: ? [L, C]

Action: ? $M_{Exit} + 2_{Action} + L_{Entry} + B_{Exit} + C_{Entry}$

Since B wasn't exited previously



Lecture Slide 36



4. **Behaviour Tree:** Implement the following behavior tree with State Machine.