Part1: Theory

Outline 1:

1. Minimal state space: {position of packman, position of unvisited landmarks}

Successor Function: Change position of packman, delete landmark from list if packman position coincides with landmark

Goal Test: set of unvisited landmarks is empty

1. Minimal state space: {position of packman, list of unvisited sites, Total number of ghosts encountered }

Successor function: Change position of packman, delete landmark from list if packman position coincides with landmark, and update Ghosts encountered

Goal test: No landmark unvisited, Total Ghosts less than Gmax

1. Minimal state space: {position of all persons, position of unvisited landmarks}.

Actions: NSEW for each

Successor function: Update positions of all people, change Boolean state of landmark if visited, increment total time of travel

Plan: Commute the cost of moving each person to their nearest unvisited landmark. Make move that generates minimum cost.

Continue till all are occupied.

Goal Test: All landmarks are visited in minimum total time

1. Minimal State space:{ locations of unvisited landmarks}

Actions: NSEW for each

Successor function: update visited monuments and max travel time for a person

Plan: Commute the individual cumulative cost of moving each person to their nearest unvisited landmark. Make move that generates minimum cumulative cost of a person.

Continue till all are occupied.

Goal:

Outline 2:

Q1. State Space: {goal state, list of point accessed, path from each point}

Instead of starting with the bee, begin with the goal state, move away from the goal in all possible directions and note the squares it can touch. Each time it reaches a square, find the optimal path i.e. calculate the path given by UCS/Astar to goal.expand till all points are covered.

Now, when the bee starts, it should follow the path corresponding to its starting position.

Heuristic can be Manhattan distance as it is the distance travelled in an ideal case (no barriers) thus the cost (tiles visited to reach goal) from the point will always be more than the suggested heuristic.

Q2.

1. UFS always give optimal results as it visits multiple paths of same cost and terminates when goal is achieved. As UFS incrementally moves from cheapest to expensive paths, if the goal state can be achieved at multiple costs/paths, UFS will return the cheapest as it will encounter it first.

BFS on the other hand may give sub optimal results in Tree search but in graph search, since it never revisits a node, it will reach each node in the cheapest possible path first. Thus it will give optimal results in graph search but not in tree search.

1. As the heuristic is admissible, we can assume for every node Xi leading from start state to S that

h (Xi)< min(cost of going from Xi to S)

h (Xi)-h(Xi+1)<cost of going from Xi to Xi+1

Thus h(Xi+1)<min(cost of going from Xi to Xi+1) + min(cost of going from Xi to S)

Thus Xi+1 is expanded first only if and only it satisfies the above criteria i.e. Xi+1 leads to path of minimum cost i.e. optimal

1. To use graph search with only admissible heuristics, we must ensure that for all nodes Xi (0<i<n) where n is finite, max(h(Xi)-h(Xk))<min(cost from Xi to Xj) (0<j<n, 0<k<n)

And none of i ,j,k is equal

Thus by limiting our heuristic values we have made it consistent as well.