# Assignment 2

**Answer 1:**

1. **Tree Graph:**
   A tree is an undirected graph having nodes and edges. Two vertices of this graph have exactly one path.

2. **Adjacency List**
   Adjacency list is a collection of unordered list that is used to represent a finite graph.

3. **Spanning Tree**
   A Spanning tree is a subset of graph G that has all the vertices of the graph covered with minimum number of edges.

4. **Breadth-first Search**
   Breadth-first search is a traversal algorithm where the traversal starts from the source node and then traverse to the neighboring nodes which are directly connected to the source and then move towards the next-level neighbor nodes.

5. **Admissible heuristic**
   Admissible heuristic is a heuristic function that never over-estimates the cost of reaching the goal.

**Answer 2:**

**Functions in the increasing order of their asymptotic growth:**

1. log n
2. √n
3. 5n
4. $n^2$√n
5. $5n^3$
6. $5n^5$
7. $0.33^n$

**Answer 3:**

**Master Theorem**

Given: T(n) = 8T (n/2) + n

x = 8, b= 2, c = 1

$\log_b a$ = 3

c < 3 therefore, the applicable case is Case 1

T(n) = n^log$_b$a
T(n) = Θ($n^3$)

**Answer 4:**

**Master Theorem**

Given: T(n) = $n^2$T (n/2) + log n
There is no applicable case as the coefficient of T() is not a constant

**Answer 5:**

**Master Theorem**

Given: T(n) = 4T (n/2) + $n^2$
A = 4, B = 2, c = 2
log$_b$a = 2
c = 2 therefore, the applicable case is Case 2
T(n) = (n^log$_b$a) $\log^{k+1}$n
T(n) = Θ($n^2$logn)

**Answer 6:**
**Given List P: 22, 13, 26, 1, 12, 27, 33, 15**

Start:
(22, 13, 26, 1, 12, 27, 33, 15)

Step 1:
Dividing the list into two parts:
P1 = (22,13,26,1)
P2 = (12,27,33,15)

Step 2:
Further dividing P1 and P2 we get:
P11 = (22,13)
P12 = (26,1)
P21 = (12,27)
P22 = (33,15)

Step 3:
Now sorting the divided parts:
P11 = (13,22)
P12 = (1,26)

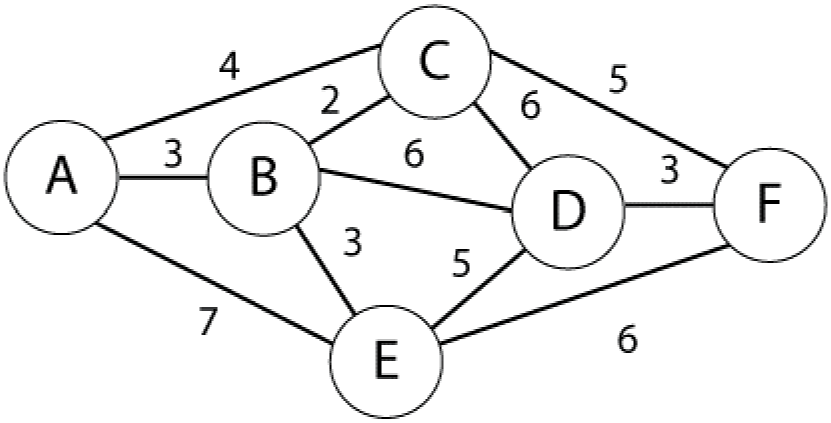
P21 = (12,27)
P22 = (15,33)

<u>Step 4</u>:
Now merging the parts in step 3:
P1 = (1, 13, 22, 26)
P2 = (12, 15, 27, 33)

<u>Step 5</u>:
Now merging the parts of step 4:
P = (1, 12, 13 ,15, 22, 26, 27, 33)

**Answer 7:**

Kruskal's Algorithm to find the minimum spanning tree



| Weights | Source | Destination |
|---|---|---|
| 2 | B | C |
| 3 | A | B |
| 3 | B | E |
| 3 | D | F |
| 4 | A | C |
| 5 | C | F |
| 5 | E | D |
| 6 | B | D |
| 6 | C | D |
| 6 | E | F |
| 7 | A | E |

Steps:

1. Connect B-C
2. Connect A-B
3. Connect B-E
4. Connect D-F
5. Skip A-C
6. Connect C-F/ Connect E-F

Stop MST has been formed with 6 nodes and 5 edges

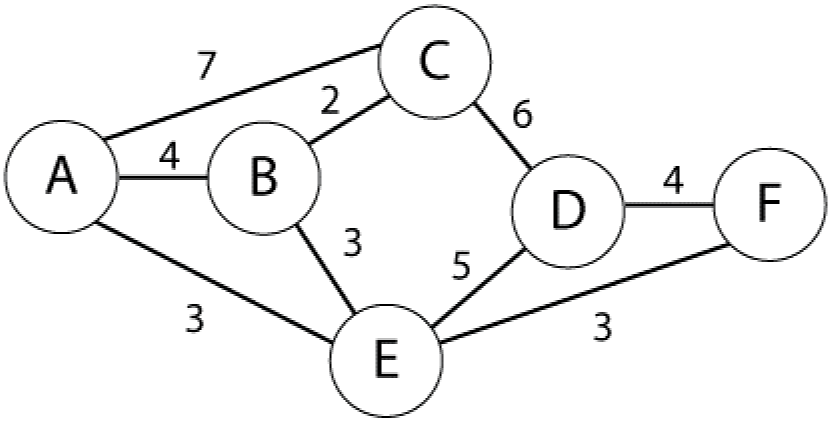
MST = (A-B, B-C, B-E, C-F, D-F)

= (A-B, B-C, B-E, E-D, D-F)

MST Weights = 2+3+3+5+3 = 16

**Answer 8:**

Prim's Algorithm to find Minimum Spanning Tree



Take the minimum edge of the cut-set each time.

0: A, S = {A}

1: A-E (3) is min-cut take A-E, S = {A, E}

2: E-B (3) is min-cut take E-B, S = {A, E, B}

3: B-C (2) is min-cut take B-C, S = {A, E, B, C}

4: E-F (3) is min-cut take E-F, S = {A, E, B, C, F}

5: F-D (4) is min-cut take F-D, S = {A, E, B, C, F, D}

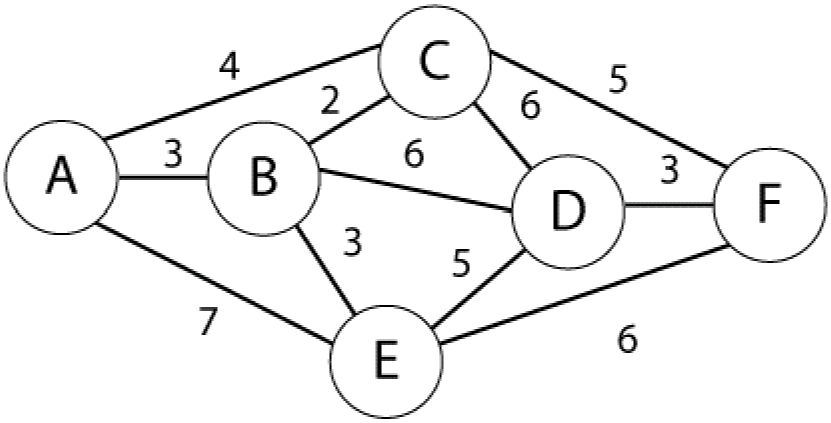
Stop MST has been formed with 6 nodes and 5 edges

MST = (A-E, E-B, B-C, E-F, F-D)

MST Weights = 3+3+2+3+4 = 15

## Answer 9:

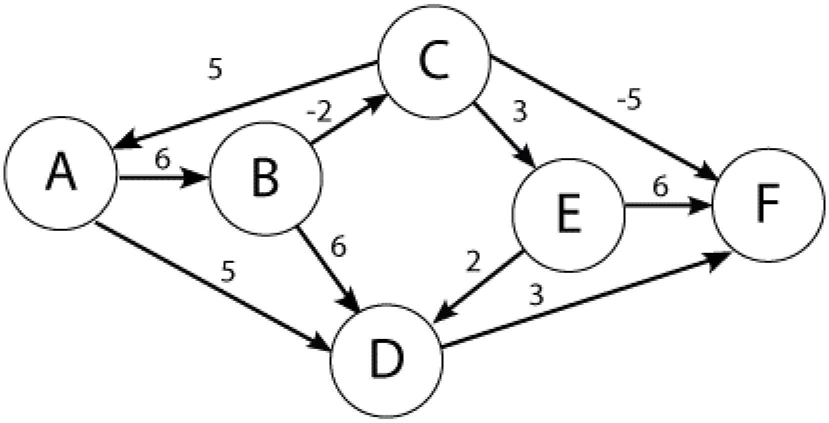
**Finding the shortest path using Dijkstra's algorithm.**



| Source A | | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|
| A {A} | A | 0 | (3, A) | (4, A) | INF | (7, A) | INF |
| B {A, B} | B | 0 | (3, A) | (5, B) | (9, B) | (6, B) | INF |
| C {A, B, C} | C | 0 | (3, A) | (5, B) | (9, B) | (6, B) | (9, C) |
| E {A, B, C, E} | E | 0 | (3, A) | (5, B) | (9, B) | (6, B) | (9, C) |
| D {A, B, C, E, D} | D | 0 | (3, A) | (5, B) | (9, B) | (6, B) | (9, C) |
| F {A, B, C, E, D, F} | F | 0 | (3, A) | (5, B) | (9, B) | (6, B) | (9, C) |

Now we return our final shortest path, which is: A → C → F Cost = 4+5 = 9

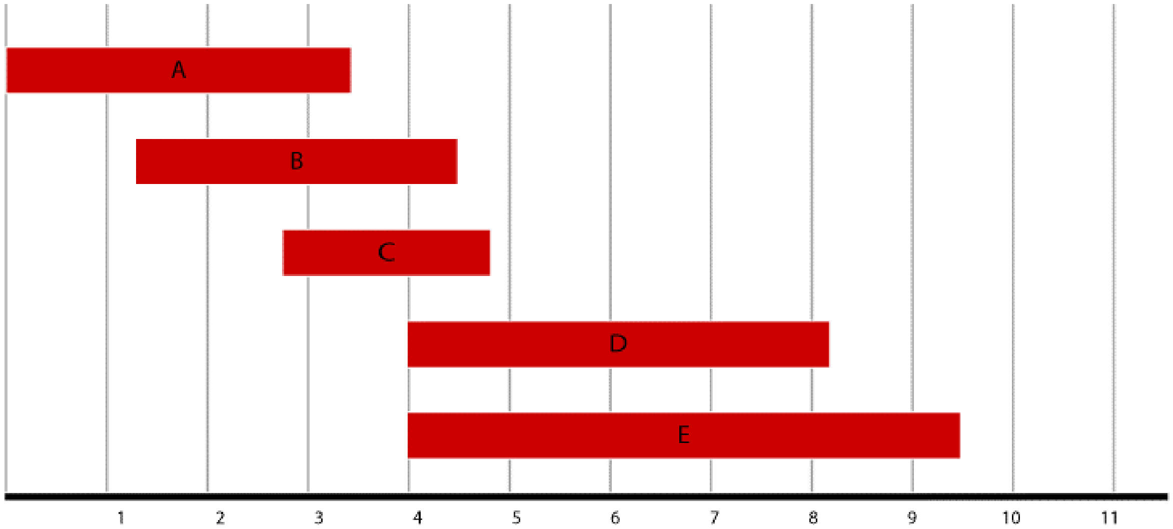
**Answer 10:**



| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 0 | 0 | INF | INF | INF | INF | INF |
| 1 | 0 | 6 | INF | 5 | INF | INF |
| 2 | 0 | 6 | 4 | 5 | 7 | 13 |
| 3 | 0 | 6 | 4 | 5 | 7 | 8 |
| 4 | 0 | 6 | 4 | 5 | 7 | -1 |

The shortest path from A – F is A → B → C → F = -1

**Answer 11:**

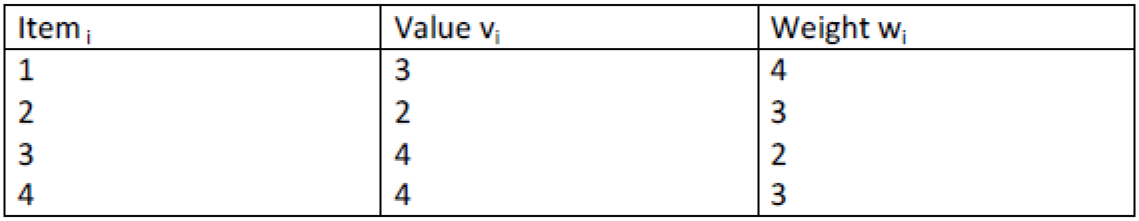


| Interval | Value |
|---|---|
| A | 2 |
| B | 3 |
| C | 2 |
| D | 3 |
| E | 2 |

| Interval | Value | Previous | Max |
|---|---|---|---|
| A | 2 | N/A | Max (4 ,0) = 4 |
| B | 3 | N/A | Max (5, 0) = 5 |
| C | 2 | N/A | Max (5, 0) = 5 |
| D | 3 | A | Max (9, 3+9) = 12 |
| E | 2 | B | Max (10, 2+10) = 12 |

S = {A, D} or {E, B}

**Answer 12: Knapsack**



| Item $_i$ | Value $v_i$ | Weight $w_i$ |
|---|---|---|
| 1 | 3 | 4 |
| 2 | 2 | 3 |
| 3 | 4 | 2 |
| 4 | 4 | 3 |

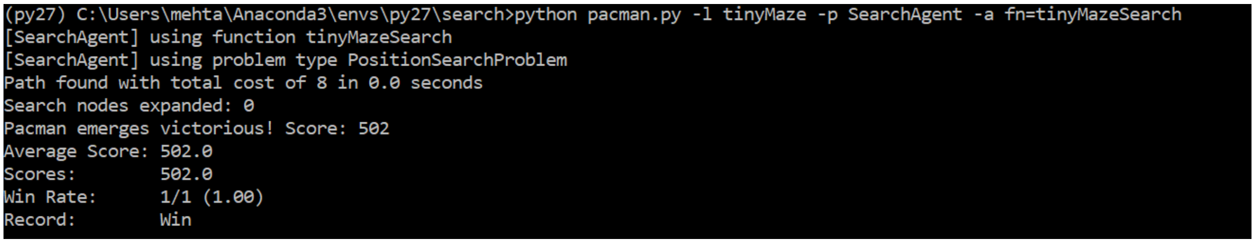
Capacity of knapsack W=6

| **Item** | **Value(V)** | **Weight(W)** | **0** | **1** | **2** | **3** | **4** | **5** | **6** |
|---|---|---|---|---|---|---|---|---|---|
| **3** | **4** | **2** | 0 | 0 | 4 | 4 | 4 | 4 | 4 |
| **4** | **4** | **3** | 0 | 0 | 4 | 4 | 4 | 8 | 8 |
| **2** | **2** | **3** | 0 | 0 | 4 | 4 | 4 | 8 | 8 |
| **1** | **3** | **4** | 0 | 0 | 4 | 4 | 4 | 4 | 7 |

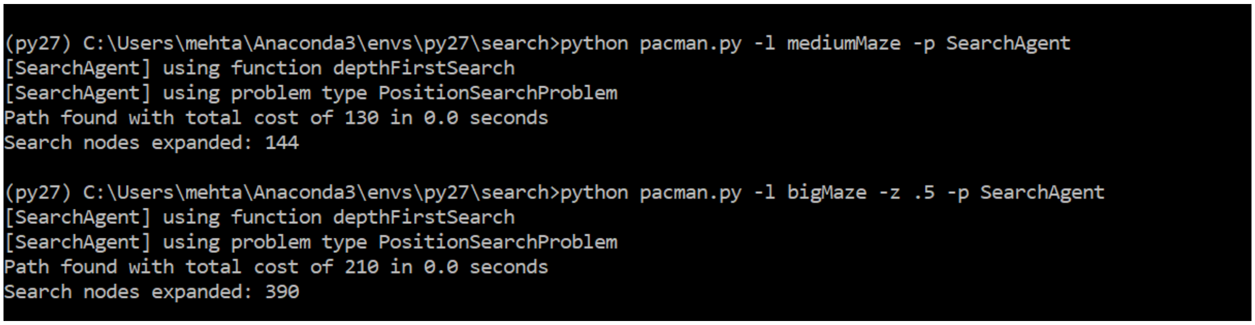
We used items 3 and 4 for a combined value of 8 in the knapsack.

S = {3,4}

## Answer 12-1

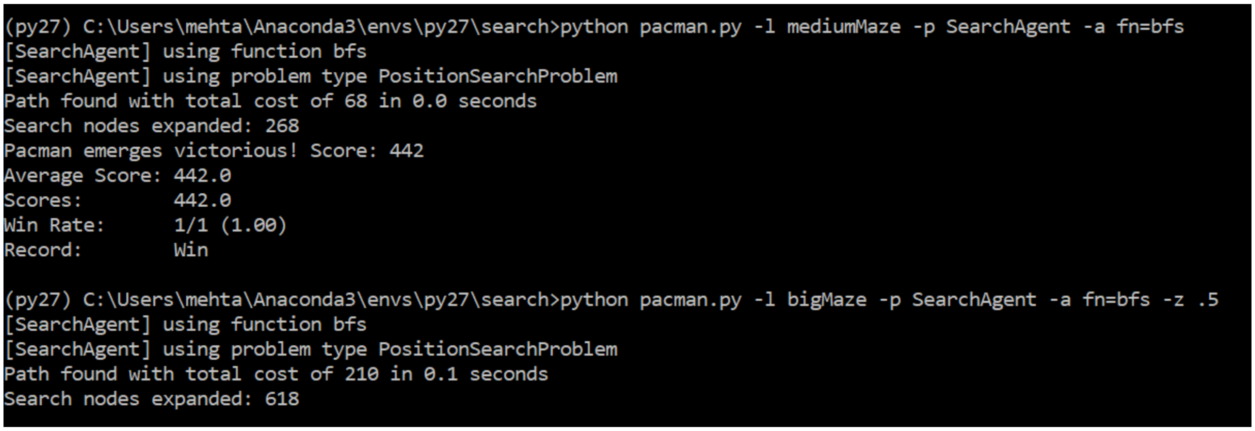


```
(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l tinyMaze -p SearchAgent -a fn=tinyMazeSearch
[SearchAgent] using function tinyMazeSearch
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 8 in 0.0 seconds
Search nodes expanded: 0
Pacman emerges victorious! Score: 502
Average Score: 502.0
Scores:        502.0
Win Rate:      1/1 (1.00)
Record:        Win
```



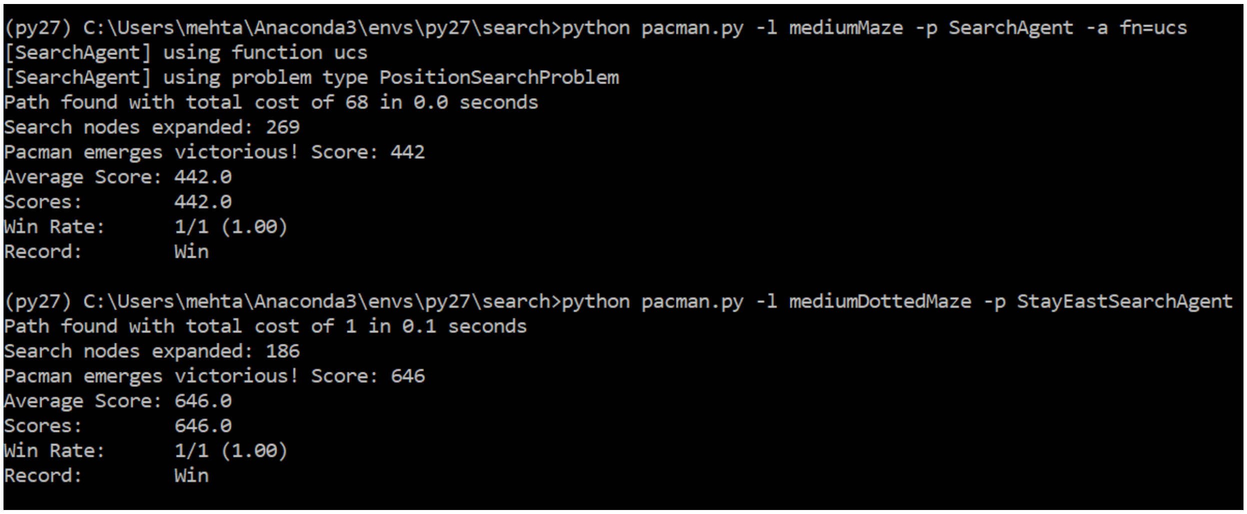
```
(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l mediumMaze -p SearchAgent
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 130 in 0.0 seconds
Search nodes expanded: 144

(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l bigMaze -z .5 -p SearchAgent
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 210 in 0.0 seconds
Search nodes expanded: 390
```

## Answer 12-2

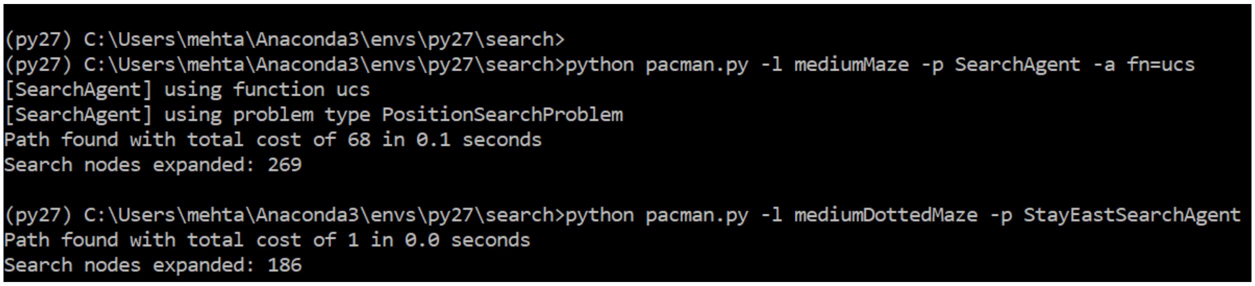


```
(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l mediumMaze -p SearchAgent -a fn=bfs
[SearchAgent] using function bfs
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 68 in 0.0 seconds
Search nodes expanded: 268
Pacman emerges victorious! Score: 442
Average Score: 442.0
Scores:        442.0
Win Rate:      1/1 (1.00)
Record:        Win

(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l bigMaze -p SearchAgent -a fn=bfs -z .5
[SearchAgent] using function bfs
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 210 in 0.1 seconds
Search nodes expanded: 618
```

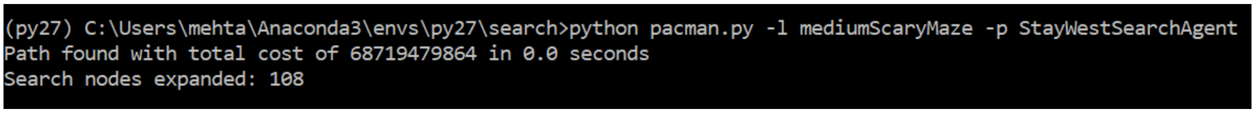
## Answer 12-3



```
(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
[SearchAgent] using function ucs
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 68 in 0.0 seconds
Search nodes expanded: 269
Pacman emerges victorious! Score: 442
Average Score: 442.0
Scores:        442.0
Win Rate:      1/1 (1.00)
Record:        Win

(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l mediumDottedMaze -p StayEastSearchAgent
Path found with total cost of 1 in 0.1 seconds
Search nodes expanded: 186
Pacman emerges victorious! Score: 646
Average Score: 646.0
Scores:        646.0
Win Rate:      1/1 (1.00)
Record:        Win
```

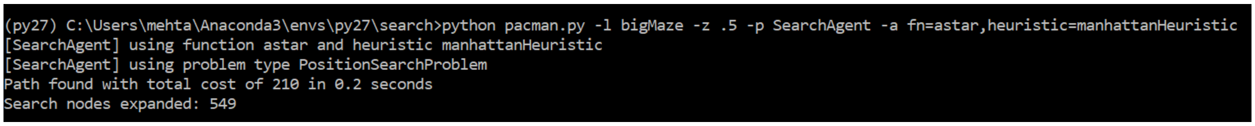


```
(py27) C:\Users\mehta\Anaconda3\envs\py27\search>
(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
[SearchAgent] using function ucs
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 68 in 0.1 seconds
Search nodes expanded: 269

(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l mediumDottedMaze -p StayEastSearchAgent
Path found with total cost of 1 in 0.0 seconds
Search nodes expanded: 186
```



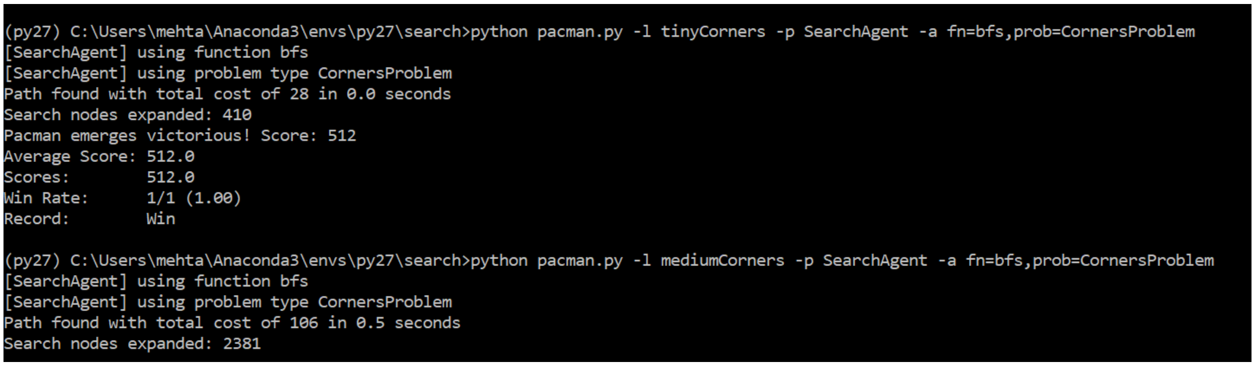
```
(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l mediumScaryMaze -p StayWestSearchAgent
Path found with total cost of 68719479864 in 0.0 seconds
Search nodes expanded: 108
```

## Answer 12-4



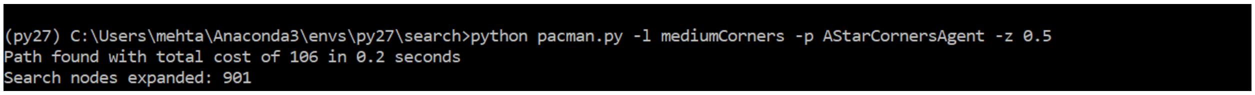
```
(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic
[SearchAgent] using function astar and heuristic manhattanHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 210 in 0.2 seconds
Search nodes expanded: 549
```

## Answer 12-5



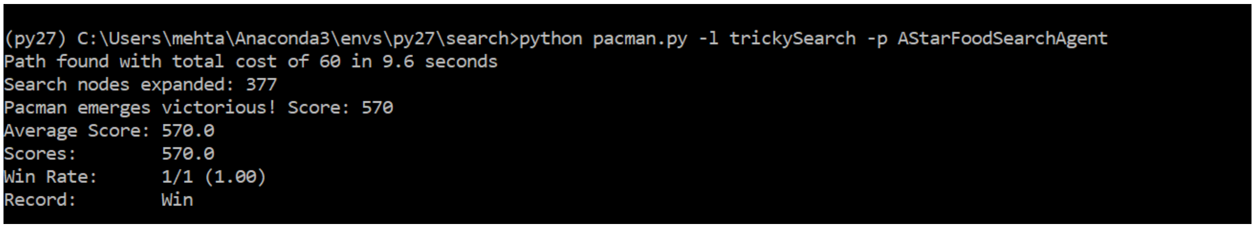
```
(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l tinyCorners -p SearchAgent -a fn=bfs,prob=CornersProblem
[SearchAgent] using function bfs
[SearchAgent] using problem type CornersProblem
Path found with total cost of 28 in 0.0 seconds
Search nodes expanded: 410
Pacman emerges victorious! Score: 512
Average Score: 512.0
Scores:        512.0
Win Rate:      1/1 (1.00)
Record:        Win

(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l mediumCorners -p SearchAgent -a fn=bfs,prob=CornersProblem
[SearchAgent] using function bfs
[SearchAgent] using problem type CornersProblem
Path found with total cost of 106 in 0.5 seconds
Search nodes expanded: 2381
```

## Answer 12-6

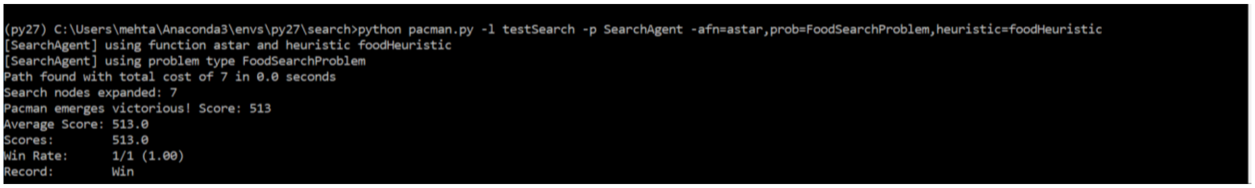


```
(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l mediumCorners -p AStarCornersAgent -z 0.5
Path found with total cost of 106 in 0.2 seconds
Search nodes expanded: 901
```

## Answer 12-7

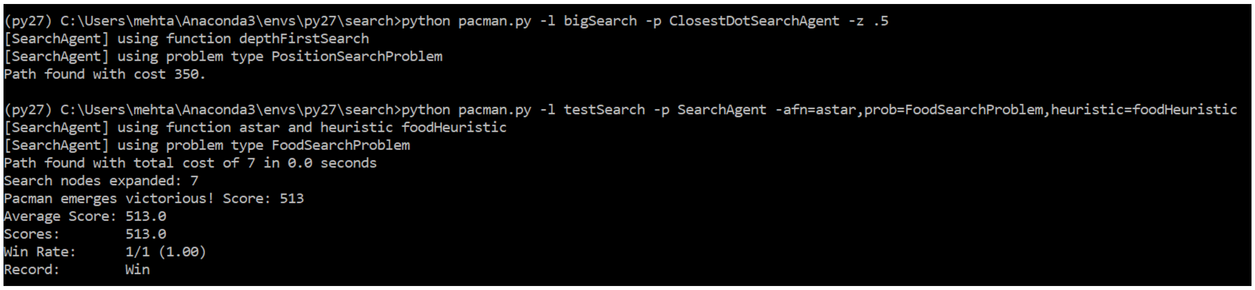


```
(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l trickySearch -p AStarFoodSearchAgent
Path found with total cost of 60 in 9.6 seconds
Search nodes expanded: 377
Pacman emerges victorious! Score: 570
Average Score: 570.0
Scores:        570.0
Win Rate:      1/1 (1.00)
Record:        Win
```



```
(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l testSearch -p SearchAgent -afn=astar,prob=FoodSearchProblem,heuristic=foodHeuristic
[SearchAgent] using function astar and heuristic foodHeuristic
[SearchAgent] using problem type FoodSearchProblem
Path found with total cost of 7 in 0.0 seconds
Search nodes expanded: 7
Pacman emerges victorious! Score: 513
Average Score: 513.0
Scores:        513.0
Win Rate:      1/1 (1.00)
Record:        Win
```

## Answer 12-8



```
(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l bigSearch -p ClosestDotSearchAgent -z .5
[SearchAgent] using function depthFirstSearch
[SearchAgent] using problem type PositionSearchProblem
Path found with cost 350.

(py27) C:\Users\mehta\Anaconda3\envs\py27\search>python pacman.py -l testSearch -p SearchAgent -afn=astar,prob=FoodSearchProblem,heuristic=foodHeuristic
[SearchAgent] using function astar and heuristic foodHeuristic
[SearchAgent] using problem type FoodSearchProblem
Path found with total cost of 7 in 0.0 seconds
Search nodes expanded: 7
Pacman emerges victorious! Score: 513
Average Score: 513.0
Scores:        513.0
Win Rate:      1/1 (1.00)
Record:        Win
```