**Assignment 2**

Q1. Give brief description of the following:

1. **Tree Graph:**

* It is an undirected graph who’s any two vertices are connected by exactly one path
* It is also known as an acyclic connected graph
* They belong to the simplest class of graphs
* Tree graphs represent a hierarchical structure in a graphical form
* They provide a range of useful applications as simple as a family tree to as complex as trees in data structures of computer science

1. **Adjacency List:**

* It is a collection if unordered lists used to represent a finite graph
* Each list describes the set of neighbors of a vertex in the graph
* It is a more space efficient way to implement a sparsely connected graph
* The advantage of the adjacency list implementation is that it allows us to compactly represent a sparse graph
* It also allows us to easily find all the links that are directly connected to a particular vertex

1. **Spanning Tree:**

* It is a subset of all the graphs G, that has all the vertices covered with a minimum possible number of edges
* A spanning tree does not have cycles and it cannot be disconnected
* All possible spanning tree have the same number of edges and vertices
* Removing one edge of the spanning tree will make it minimally connected
* Adding one edge to the spanning tree will create a circuit/loop making it maximally acyclic

1. **Breadth-first Search (BFS):**

* It is an algorithm for traversing or searching tree or graph data structures
* It starts at the tree root and explores the neighbor nodes first, before moving to the next level neighbors
* This algorithm is used to solve many problems including finding the shortest path in a graph and solving puzzle games

Eg – Rubik’s Cube

* BFS consists of three types of vertices namely tree vertices, fringe vertices and undiscovered vertices

1. **Admissible Heuristic:**

* A heuristic function is said to be admissible if it never overestimates the cost of reaching the goal
* A heuristic is specific to a particular state space, and also to a particular goal state in that state space
* It must be admissible to all the states in that search space

Q2. Arrange the following functions in the increasing order of asymptotic growth:

Solution: 0.33n

log n

√n

n2 √n

5n3

5n5

5n

Q3. Master Theorem: For the following recurrence, give an expression for the runtime T(n) if the recurrence can be solved with the Master Theorem. Otherwise, indicate that the Master Theorem does not apply.

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

**Solution:** k = logba

= log28

= log(8)/log(2)

= 3

f(n) = n which is θ(n3)

Now, θ(nk) = θ(nlogba)

Therefore, the runtime T(n) is θ(n3)

Q4. Master Theorem: For the following recurrence, give an expression for the runtime T(n) if the recurrence can be solved with the Master Theorem. Otherwise, indicate that the Master Theorem does not apply.

T(n) = n2 T (n/2) + log n

**Solution:** Theorem is not applicable

Q5. Master Theorem: For the following recurrence, give an expression for the runtime T(n) if the recurrence can be solved with the Master Theorem. Otherwise, indicate that the Master Theorem does not apply.

T(n) = 4T (n/2)+ n2

**Solution:** Theorem does not apply (Polynomial difference)

Q6. Sort the list of integers below using Merge sort. Show your work. Write a recurrence relation for Merge sort.

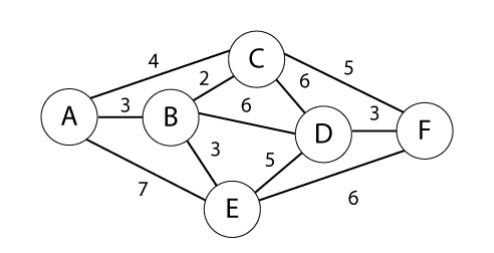
(22, 13, 26, 1, 12, 27, 33, 15)

**Solution:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Original | 22 | 13 | 26 | 1 | 12 | 27 | 33 | 15 |  |  |  |  |  |  |  |
| Divide in 2 | 22 | 13 | 26 | 1 |  | 12 | 27 | 33 | 15 |  |  |  |  |  |  |
| Divide in 4 | 22 | 13 |  | 26 | 1 |  | 12 | 27 |  | 33 | 15 |  |  |  |  |
| Divide in 8 | 22 |  | 13 |  | 26 |  | 1 |  | 12 |  | 27 |  | 33 |  | 15 |
| Merge 1 | 13 | 22 |  | 1 | 26 |  | 12 | 27 |  | 15 | 33 |  |  |  |  |
| Merge 2 | 1 | 13 | 22 | 26 |  |  | 12 | 15 | 27 | 33 |  |  |  |  |  |
| Merge 3 | 1 | 12 | 13 | 15 | 22 | 26 | 27 | 33 |  |  |  |  |  |  |  |

Recurrence Relation: T(n) = 2 T(n/2) + n

Q7. Use Kruskal's algorithm to find a minimum spanning tree for the connected weighted graph below:



Steps:

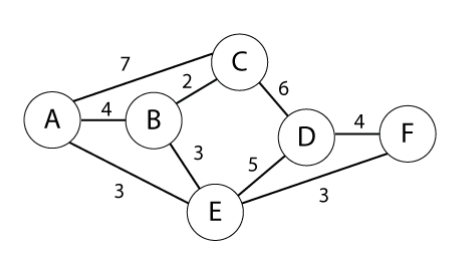
1. Connect A-B(3)
2. Connect B-C(2)
3. Connect B-E(3)
4. Connect C-F(5)
5. Connect D-F(3)

MST formed (5 edges, 5 vertices)

MST = {A-B, B-C, B-E, C-F, D-F}

MST weight = 3+2+3+5+3 = 16

Q8. Use Prim's algorithm to find a minimum spanning tree for the connected weighted graph below. Show your work.



**Solution:**

Steps: S = {A}

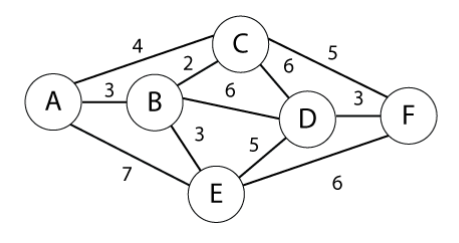
1. A-E (3) is min-cut take A-E S={A, E}
2. A-B (4) is min-cut take A-B S={A, E, B}
3. B-C (2) is min-cut take B-C S={A, E, B, C}
4. E-F (3) is min-cut take E-F S={A, E, B, C, F}
5. F-D (4) is min-cut take F-D S={A, E, B, C, F, D}

Here we are done with n-1 edges. Hence we stop.

MST = {A-E, A-B, B-C, E-F, F-D}

MST weight = 3+4+2+3+4 = 16

Q9. Find shortest path from A to F in the graph below using Dijkstra's algorithm. Show your steps.



**Solution:**

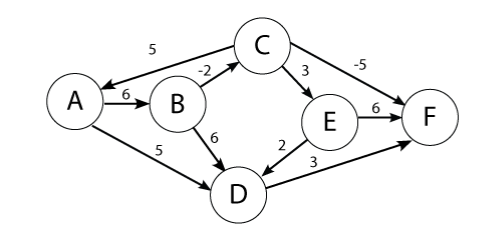
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | A | B | C | D | E | F |
| A{A} | A | 0 | (3,A)\*\* | (4,A) | INF | (7,A) | INF |
| B{A,B} | B | 0 | (3,A) | (5,B)\*\* | (9,B) | (6,B) | INF |
| C{A,B,C} | C | 0 | (3,A) | (5,B) | (2,C) | (6,B)\*\* | (11,C) |
| E{A,B,C,E} | E | 0 | (3,A) | (5,B) | (11,E)\*\* | (6,B) | (12,E) |
| D{A,B,C,E,D} | D | 0 | (3,A) | (5,B) | (11,E) | (6,B) | (12,E)\*\* |
| F{A,B,C,E,D,F} | F | 0 | (3,A) | (5,B) | (11,E) | (6,B) | (12,E) |

PS: \*\* indicates the shortest path

Final shortest path is A 🡪 B 🡪 E 🡪 F

Cost = 3+3+6 = 12

Q10. Use the Bellman-Ford algorithm to find the shortest path from node A to F in the weighted directed graph above. Show your work.

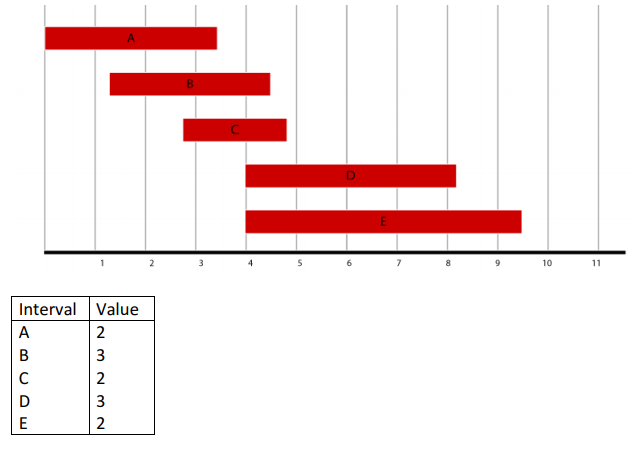


**Solution:**

Shortest path: A 🡪 B 🡪 C 🡪 F at cost -1

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

Q11. Given the five intervals below, and their associated values; select a subset of non overlapping intervals with the maximum combined value. Use dynamic programming. Show your work.

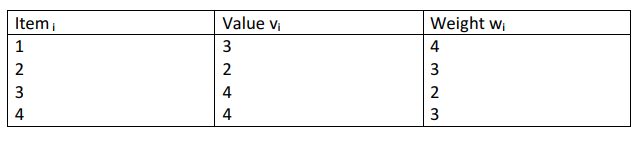


**Solution:**

|  |  |  |  |
| --- | --- | --- | --- |
| Interval | Value | Previous | Max |
| A  B  C  D  E | 2  3  2  3  2 | n/a  n/a  n/a  A  A | Max(2, 0) = 2  Max(3, 2) = 3  Max(2, 3) = 3  Max(3, 2+3) = 5  Max(5, 2+2) = 5 |
|  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Interval | Trace(i) | S |
| E | 2 + 2 < 5 | {} |
| D | 2+3>=5, jump to C | {D} |
| C | jump to A | {D} |
| B | jump to A |  |
| A | 2+0> = 2 | {D,A} |

Q12. Given the weights and values of the four items in the table below, select a subset of items with the maximum combined value that will fit in a knapsack with a weight limit, W, of 6. Use dynamic programming. Show your work.



Capacity of knapsack W=6

**Solution:**

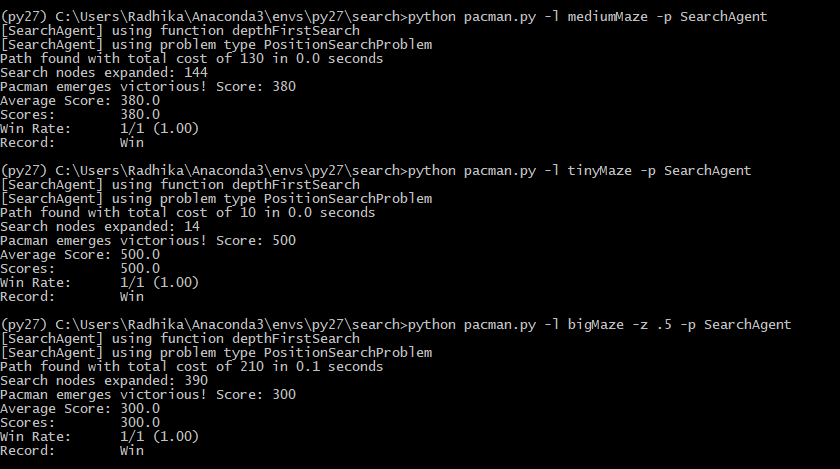
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **6** | 3 | 3 | 7 | 8 <-- |
| **5** | 3 | 3 | 6 | 8 |
| **4** | 3 | 3 | 4 | 4 |
| **3** | 0 | 2 | 4 <-- | 4 |
| **2** | 0 | 0 | 4 | 4 |
| **1** | 0 <-- | 0 <-- | 0 | 0 |
|  | **1** | **2** | **3** | **4** |

Items 3 and 4 are being used for a combined value of 8 in the knapsack.

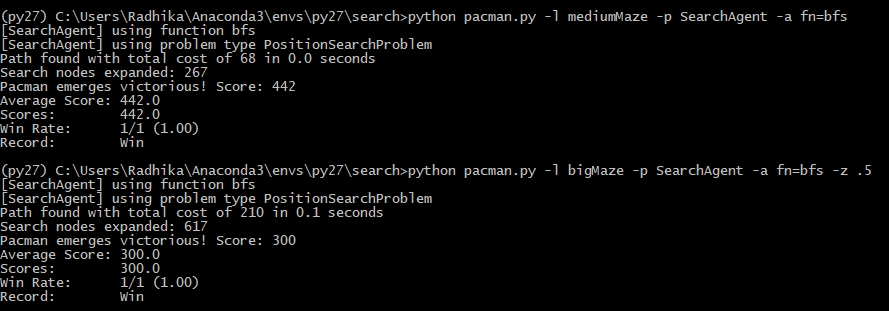
S = {3,4}

Q13. Search in Pacman

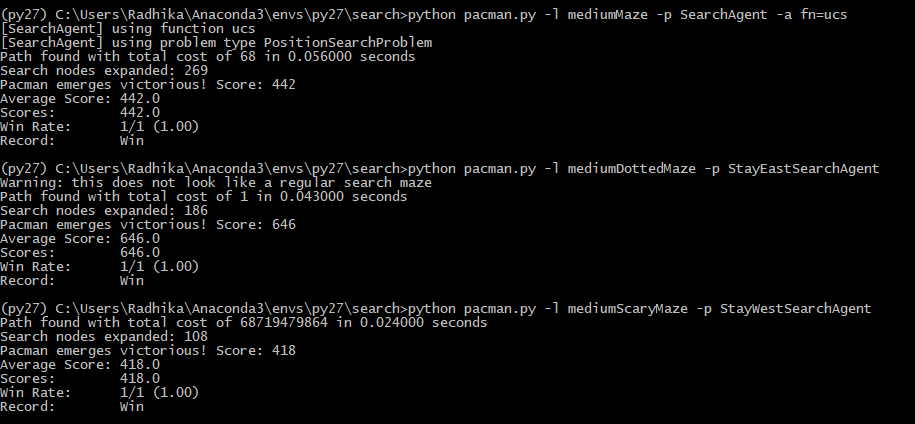
1. Finding a Fixed Food Dot using Depth First Search



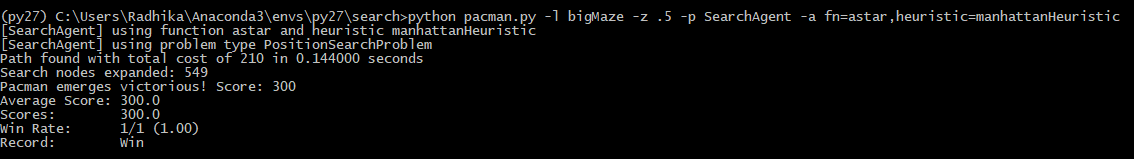
1. Breadth First Search



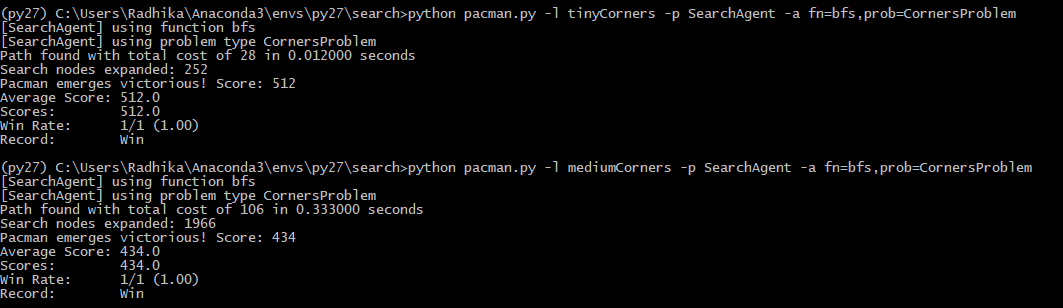
1. Varying the Cost Function



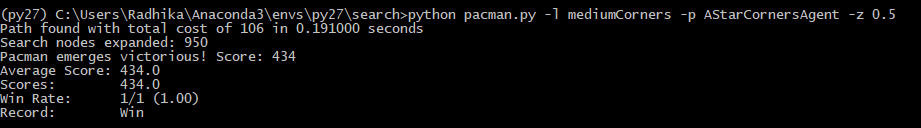
1. A\* search



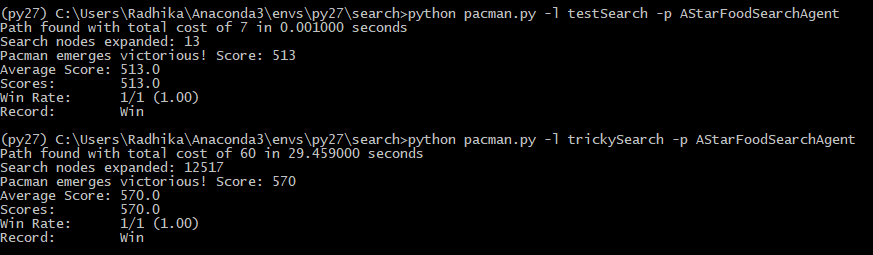
1. Finding All the Corners



1. Corners Problem: Heuristic



1. Eating All The Dots



1. Suboptimal Search

