**Big-Data Systems and Intelligence Analytics**

**CSYE 7245 - SPRING 2018**

Assignment 2

Question 1: Give a brief definition for the following:

i. Tree graph

ii. Adjacency List

iii. Spanning Tree

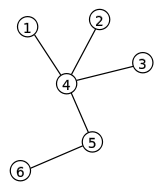
iv. Breadth-first search (BFS)

v. Admissible heuristic

Solution:

i. Tree graph:

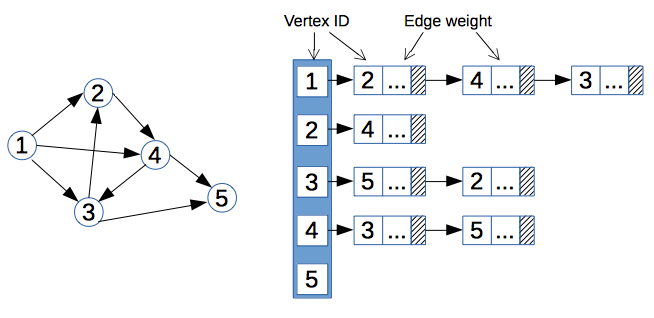
An acyclic connected graph consisting of nodes (vertices) and edges i.e. an undirected graph where two vertices are connected with exactly one path.



A labeled tree with 6 vertices and 5 edges.

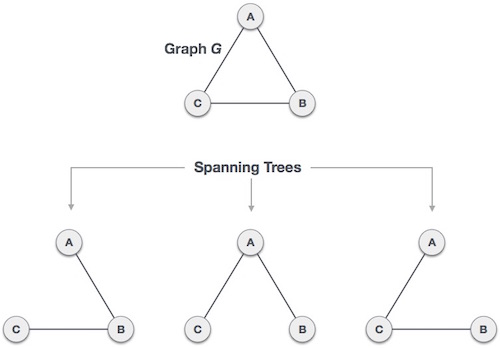
ii. Adjacency List:

An adjacency list is a collection of unordered lists used to represent a finite [graph](https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)). Each list describes the set of neighbors of a [vertex](https://en.wikipedia.org/wiki/Vertex_(graph_theory)) in the graph.



iii. Spanning Tree:

A spanning tree is a subset of Graph G, which has all the vertices covered with minimum possible number of edges. Hence, a spanning tree does not have cycles and it cannot be disconnected. A complete undirected graph can have maximum nn-2 number of spanning trees, where n is the number of nodes.



iv. Breadth-first search (BFS):

Breadth-first search (BFS) is an [algorithm](https://en.wikipedia.org/wiki/Algorithm) for traversing or searching [tree](https://en.wikipedia.org/wiki/Tree_data_structure) or [graph](https://en.wikipedia.org/wiki/Graph_(data_structure)) data structures. It starts at the [tree root](https://en.wikipedia.org/wiki/Tree_(data_structure)#Terminology) and explores the neighbor nodes first, before moving to the next level neighbors.



Order in which the nodes are expanded

v. Admissible heuristic:

In [algorithms](https://en.wikipedia.org/wiki/Algorithm) related to [pathfinding](https://en.wikipedia.org/wiki/Pathfinding), a [heuristic function](https://en.wikipedia.org/wiki/Heuristic_function) is said to be admissible if it never overestimates the cost of reaching the goal, i.e. the cost it estimates to reach the goal is not higher than the lowest possible cost from the current point in the path.

Question 2: Arrange the following functions in increasing order of asymptotic growth:

• 5n5

• 0.33n

• 5n3

• n2 √n

• 5n

• log n

• √n

Solution:

1. 0.33n (Exponentially decreasing)
2. log n
3. √n
4. n2 √n
5. 5n3
6. 5n5
7. 5n (Exponentially increasing)

Question 3: 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:

For T(n) = 8T (n/2) + n, a=8, b=2, d=1

As logba > d

=> log28 = 3 > 1

We can say that T(n)= Theta (nd) = Theta(n1)

T(n)= Theta (n)

Question 4: 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) = n2T (n/2) + log n

Solution:

Master Theorem does not apply to T(n) = n2T (n/2) + log n as A is not a constant

Question 5: 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:

For T(n) = 4T (n/2) + n2 , a=4, b=2, d=2

As logba = d

=> log24 = 2 = d

We can say that T(n)= Theta (ndlog (n)) = Theta(n2log(n))

T(n)= Theta (n2log (n))

Question 6: 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

Question 7: Use Kruskal's algorithm to find a minimum spanning tree for the connected

weighted graph below:



Solution:

* Connect B-C (2)
* Connect A-B (3)
* Connect B-E (3)
* Connect D-F (3)
* Skip A-C cycle (4)
* Connect E-D (5)
* Stop. Minimum spanning tree formed with 5 edges and 6 vertices.

Question 8: Use Prim's algorithm to find a minimum spanning tree for the connected

weighted graph below. Show your work.



Solution:

* A S={A}
* A-E (3) < A-B (4) or A-C (7) Take A-E S= {A, E}
* E-B (3) < A-B (4) or A-C (7) or E-D (5) Take E-B S= {A, E, B}
* E-F (3) < A-B (4) or A-C (7) or E-D (5) Take E-F S= {A, E, B, F}
* B-C (2) < A-B (4) or A-C (7) or D-C (6) Take B-C S= {A, E, B, F, C}
* F-D (4) < A-B (4) or A-C (7) or D-C (6) or E-D (5) Take F-D S= {A, E, B, F, C, D}

Done (n-1) edges.

Maximum Spanning Tree= {A-E, E-B, E-F, B-C, F-D}

Question 9: Find shortest path from A to F in the graph below using Dijkstra's algorithm. Show

your steps.



Solution:

Source A

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | A | B | C | D | E | F |
| 1: A {A} | A | 0 | (3, A) \*\* | (4, A) | INF | (7, A) | INF |
| 2: B {A, B} | B | 0 | (3, A) | (4, A) \*\* | (9, B) | (6, B) | INF |
| 3: C {A, B, C} | C | 0 | (3, A) | (4, A) | (9, B) | (6, B) \*\* | (9, C) |
| 4: E {A, B, C, E} | E | 0 | (3, A) | (4, A) | (9, B) \*\* | (6, B) | (9, C) |
| 5: D {A, B, C, E, D} | D | 0 | (3, A) | (4, A) | (9, B) | (6, B) | (9, C) \*\* |
| 4: F {A, B, C, E, D, F} | F | 0 | (3, A) | (4, A) | (9, B) | (6, B) | (9, C) |

Question 10: 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:

Edges and their respective weights are as follows:

|  |  |
| --- | --- |
| **Edges** | **Weights** |
| AB | 6 |
| AD | 5 |
| BC | -2 |
| BD | 6 |
| CA | 5 |
| CE | 3 |
| CF | -5 |
| DF | 3 |
| ED | 2 |
| EF | 6 |

There will be a total of (n-1) iterations where ‘n’: number of vertices

Initial: A (0) B (INF) C (INF) D (INF) E (INF) F (INF)

Iteration 1: A (0) B (6) C (4) D (5) E (7) F (-1)

Iteration 2: A (0) B (6) C (4) D (5) E (7) F (-1)

Since there are no changes in both iterations we can stop.

Question 11: Given the five intervals below, and their associated values; select a subset of nonoverlapping

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 | NA  NA  NA  A  A | Max (0, 2) = 2  Max (2, 3) = 3  Max (3, 2) = 3  Max (3, 3+2) = 5  Max (5, 2+2) = 5 |

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

S= {D, A}

Question 12: 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.

|  |  |  |
| --- | --- | --- |
| **Item** | **Value** | **Weight** |
| 1 | 3 | 4 |
| 2 | 2 | 3 |
| 3 | 4 | 2 |
| 4 | 4 | 3 |

Solution:

Capacity of knapsack W=6

Algorithm: given two arrays w[4, 3, 2, 3] and v[3, 2, 4, 4]:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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** |

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

S={3,4}