CSYE 7245 Big Data Systems & Intelligence Analytics

Assignment 2

Emily Strong

February 28, 2018

**Q1:** Give a brief definition for the following:

1. Tree graph – A tree graph is an acyclic connected graph. All vertices in the tree are connected by one path with no cycles.
2. Adjacency List – An adjacency list is a collection of lists in which each list represents the vertices adjacent to a given vertex, as well as the edge weights if it is a weighted graph. It is implemented as an array of linked lists.
3. Spanning Tree – A spanning tree is a tree that connects all of the vertices in a graph using some or all of the edges.
4. Breadth-first search (BFS) – BFS is a graph traversal algorithm in which, starting at the root, each of the neighbors of a given vertex are explored, then the neighbors of its sibling vertices are explored before proceeding to the next level neighbors. That is, each level of the tree is explored in succession rather than each branch.
5. Admissible heuristic – An admissible heuristic is one that always estimates a cost that is lower than or equal to the actual cost – this is, it never overestimates the cost.

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

* 5n5
* 0.33n
* 5n3
* n2 √n
* 5n
* log n
* √n

**Answer**:

1. 0.33n is exponentially decreasing with Θ(1)
2. log n is Θ(log n)
3. √n is Θ(n1/2)
4. n2 √n is Θ(n5/2)
5. 5n3 is Θ( n3)
6. 5n5 is Θ( n5)
7. 5n is exponentially increasing with Θ(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

**Answer:**

a = 8, b = 2, log2(8) = 3

This is case 1:

F(n) = n = O(n3-2) where ε= 2

Therefore T(n) = 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) = n2T (n/2) + log n

**Answer:**

Since a = n2 and is thus not a constant, the Master Theorem does not apply.

**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

**Answer:**

A = 4, b = 2, log2(4) = 2

This is case 2:

F(n) = n2 =Θ(n2log0n) where k = 0

Therefore T(n) = Θ(n2log n)

**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)

Answer:

Step 1 divide in 2: [22, 13, 26, 1], [12, 27, 33, 15]

Step 2 divide in 4: [22, 13], [26, 1], [12, 27], [33, 15]

Step 3 divide in 8: [22], [13], [26], [1], [12], [27], [33], [15]

Step 4 merge: [13, 22], [1, 26], (12, 27], [15, 33]

Step 5 merge: [1, 13, 22, 26], [12, 15, 27, 33]

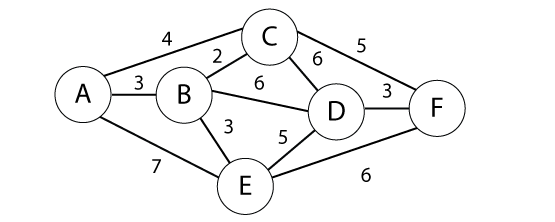
Step 6 merge: [1, 12, 13, 15, 22, 26, 27, 33]

This can also be represented in a table:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Unsorted | 22 | 13 | 26 | 1 | 12 | 27 | 33 | 15 |  |  |  |  |  |  |  |
| Step 1: Divide in 2 | 22 | 13 | 26 | 1 |  | 12 | 27 | 33 | 15 |  |  |  |  |  |  |
| Step 2:  Divide in 4 | 22 | 13 |  | 26 | 1 |  | 12 | 27 |  | 33 | 15 |  |  |  |  |
| Step 3: Divide in 8 | 22 |  | 13 |  | 26 |  | 1 |  | 12 |  | 27 |  | 33 |  | 15 |
| Step 4: Merge 1 | 13 | 22 |  | 1 | 26 |  | 12 | 27 |  | 15 | 33 |  |  |  |  |
| Step 5: Merge 2 | 1 | 13 | 22 | 26 |  | 12 | 15 | 27 | 33 |  |  |  |  |  |  |
| Step 6: Merge 3 | 1 | 12 | 13 | 15 | 22 | 26 | 27 | 33 |  |  |  |  |  |  |  |

The recurrence relation for this is that each divide step is 2 \* T(n/2) and each merge step is n. Thus T(n) = 2T(n/2) + n. Since log2(2) = 1, this is case 2 in the Master Theorem with k = 0, giving T(n) = Θ(n log n).

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



**Answer:**

Steps:

1: Connect B-C (2)

2: Connect A-B (3)

3: Connect B-E (3)

4: Connect D-F (3)

5: Skip A-C (4) because it would create a cycle

6: Connect C-F (5)

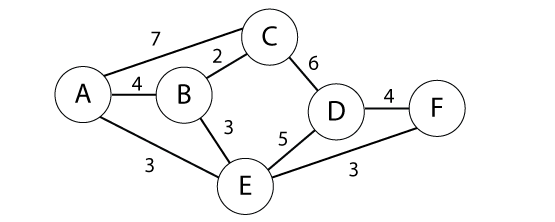
Stop. A minimum spanning tree has been formed with 6 vertices and 5 edges.

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

MST Weight = 16

Because when edges have the same weight one is chosen arbitrarily, step 6 could have instead been Connect E-D (5) which would have created a different minimum spanning tree with the same weight of 16.

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



**Answer:**

Steps:

0: Start with A, S = {A}

1: A-E (3) is the minimum-cut. Take A-E, S = {A, E}

2: B-E (3) or E-F(3) is the minimum-cut. Take B-E, S = {A, B, E}

3: B-C (2) is the minimum-cut. Take B-C, S = {A, B, C, E}

4: E-F (3) is the minimum-cut. Take E-F, S = {A, B, C, E, F}

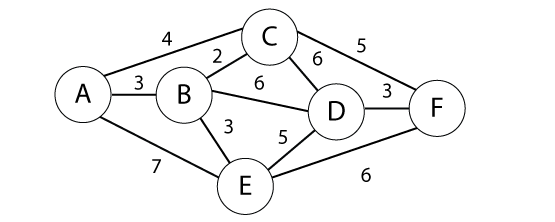
5: D-F (4) is the minimum cut. Take D-F, S = {A, B, C, D, E, F}

Done with n-1 edges (5).

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

MST Weight = 15

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



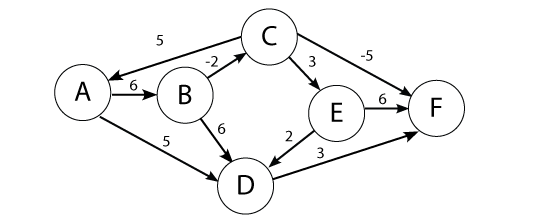
**Answer:**

\*\* indicates U with shorted path to s

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

The shorted path from A to F is A → C → F with cost 4 + 5 = 9

**Q10:**



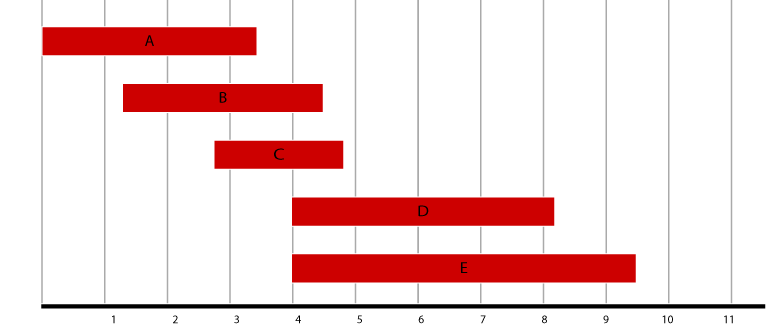
Use the Bellman-Ford algorithm to find the shortest path from node A to F in the weighted directed graph above. *Show your work.*

**Answer:**

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

The shortest path from A to F is A → B → C → F with cost 6 - 2 - 5 = -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.*



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

**Answer:**

Evaluate combinations:

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

Trace the set:

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

S = {A, D}

**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.*

|  |  |  |
| --- | --- | --- |
| Item i | Value vi | Weight wi |
| 1  2  3  4 | 3  2  4  4 | 4  3  2  3 |

Capacity of knapsack W=6

**Answer:**

Bottom-Up Values:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Weight** | **1** | **2** | **3** | **4** |
| 1 | Max(0, 0) = 0 | Max(0, 0) = 0 | Max(0, 0) = 0 | Max(0, 0) = 0 |
| 2 | Max(0, 0) = 0 | Max(0, 0) = 0 | Max(0, 4) = 4 | Max(4, 0) = 4 |
| 3 | Max(0, 0) = 0 | Max(0, 2) = 2 | Max(2, 4) = 4 | Max(4, 4) = 4 |
| 4 | Max(0, 3) = 3 | Max(3, 2) = 3 | Max(3, 4) = 4 | Max(4, 4) = 4 |
| 5 | Max(0, 3) = 3 | Max(3, 2) = 3 | Max(3, 2+4) = 6 | Max(6, 4+4) = 8 |
| 6 | Max(0, 3) = 3 | Max(3, 2) = 3 | Max(3, 3+4) = 7 | Max(7, 4+4) = 8 |

S = {3, 4} has a value of 8