

CS 5/7350 - Test#1
March, 2020

Name: Bingying Liang
ID: 48999397

1. [9 pts] Define the following Terms as succinctly as possible:

(a) Algorithm:

Solution: A step by step procedure for solving a problem in a **finite amount of time**.

(b) Tree

Solution: A connected acyclic graph. (Does not have to be rooted)

(c) In-Order Traversal

Solution: Visiting the vertices of a rooted tree by recursively visiting the left subtree, the current vertex and then the right subtree.

(d) Graph

Solution: A collection of elements (Vertices) and a relation on those elements(edges)

(e) NP-Hard

Solution: A class of problems that at least as hard (may or may not also be NP problem itself) and some are possibly harder than all NP problems. (solve 1, solve all NP problems).

(f) Insertion Sort

Solution: A method of ordering items by taking each item and placing it where it belongs among the other items already taken.

2. [8 pts] Solve the following problems:

(a) 2^{55} modulo 11 =

Solution: 10

$$2^{55} = (2^{10} \times 2^{10} \times 2^{10} \times 2^{10} \times 2^{10} \times 2^5) \bmod 11 = (1 \times 1 \times 1 \times 1 \times 1 \times 10) = 10$$

(b) **Given** $|M| = 6$ & $|N| = 5$, **find** $|\text{Power Set of}(M \times N)|$

Solution:

$$2^n = 2^{M \times N} = 2^{30}$$

(c) Number of edges in a complete graph with 10 vertices =

Solution: 45

(d) $-(1/4)$ modulo 7 =

Solution: 5

$$\left(-\frac{1}{4} \times 4\right) \bmod 7 = (-1) \bmod 7 = 6$$

$$(4 \times 5) \bmod 7 = 6$$

$$\therefore \left(-\frac{1}{4}\right) \bmod 7 = 5$$

3. [14 pts] For each pair below, circle the choice that has the higher asymptotic upper bound. If they are the same, circle "same".

Solution:

(i) $1 + 2 + 3 + 4 + \dots + n$ or $n^2 + 3n + 65$ or same

(ii) $\frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} + \dots + \frac{1}{n}$ or n or same

(iii) $1 * 2 * 3 * 4 * 5 * \dots * n$ or $1 * 2 * 3 * \dots * n * (n+1)$ or same

(iv) $3 + 3 + \dots + 3$ (n times) or $2 + 2 + \dots + 2$ (n times) or same

(v) $3 * 3 * \dots * 3$ (n times) or $2 * 2 * \dots * 2$ (n times) or same

(vi) $1 + 2 + 3 + 4 + \dots + n$ or $1 + 2 + 3 + 4 + \dots + n + (n+1)$ same

(vii) n^2 or $(n+1)^2$ or same

4. [5 pts] Argue that the problem, H, of creating a MIN-HEAP from an unsorted array of integers using the HEAPIFY algorithm discussed in class is at least as hard - and maybe even harder - than the problem, M, of finding the minimum element of the same unsorted array of integers.

Solution: Given a solution to Problem H, I can use that solution to solve problem M by simply reading the first element of the heap created by the solution to H.

5. [9 pts] Determine a Huffman encoding for each symbol in a message that contains:

8 C's, 8D's, 3 E's, 3 F's, 2G's, 1H and 1K.

- (a) How many bits are in the entire message if each symbol is encoded with 3 bits?

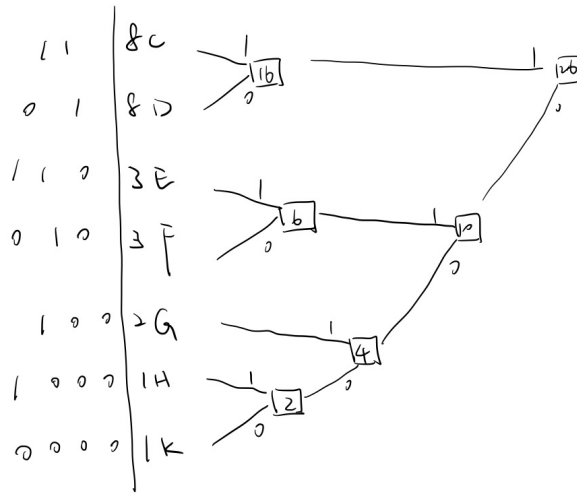
Solution:

$$(8 + 8 + 3 + 3 + 2 + 1 + 1) \times 3 = 26 \times 3 = 78 \text{ bits}$$

- (b) How many bits are in the entire Huffman coded message?
(c) How much entropy (information) is in the entire message?

Solution:

b)



$$\begin{aligned}
 & 8 \times 2 + 8 \times 2 + 3 \times 3 + 3 \times 3 + 2 \times 3 + 1 \times 4 + 1 \times 4 \\
 &= 16 + 16 + 9 + 9 + 6 + 4 + 4 \\
 &= 32 + 18 + 10 + 4 \\
 &= 50 + 14 \\
 &= 64 \text{ bits}
 \end{aligned}$$

c) $8 + 8 + 3 + 3 + 2 + 1 + 1 = 16 + 6 + 2 + 2 = 16 + 10 = 26$

$$P_C = \frac{8}{26} = \frac{4}{13} \quad P_D = \frac{4}{13} \quad P_E = \frac{3}{26} \quad P_F = \frac{3}{26} \quad P_G = \frac{1}{13} \quad P_H = \frac{1}{26} \quad P_K = \frac{1}{26}$$

$$\begin{aligned}
 & 8 \log_2 \frac{1}{P_C} + 8 \log_2 \frac{1}{P_D} + 3 \log_2 \frac{1}{P_E} + 3 \log_2 \frac{1}{P_F} + 2 \log_2 \frac{1}{P_G} + \log_2 \frac{1}{P_H} + \log_2 \frac{1}{P_K} \\
 &= 8 \log_2 \frac{13}{4} + 8 \log_2 \frac{13}{4} + 3 \log_2 \frac{26}{3} + 3 \log_2 \frac{26}{3} + 2 \log_2 13 + \log_2 26 + \log_2 26 \\
 &= 8 \times 1.70 + 8 \times 1.70 + 3 \times 3.12 + 3 \times 3.12 + 2 \times 3.70 + 1 \times 4.70 + 1 \times 4.70 \\
 &= 27.2 + 18.72 + 7.4 + 7.4 = 62.72 \text{ bits}
 \end{aligned}$$

6. [6 pts] Two people need to establish a secret key for encrypting communications. They agree to use a Diffie-Hellman key exchange with a modulus of 11 and decide on 2 as the base. Person A chooses a random value of 8 and person B chooses a random value of 9.

(a) What is the value Person A sends to Person B

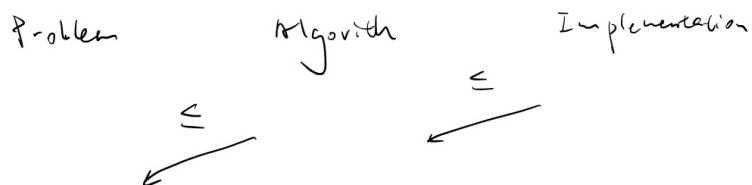
Solution: 3

Solution: 6

Solution: 4

- | Problem | | | Algorithm | | | Implementation | | |
|---------|-----|-------|---------------|---------------|---------------|----------------|-----|-------|
| Best | Avg | Worst | Best | Avg | Worst | Best | Avg | Worst |
| | | | | | $\Theta(n^2)$ | | | |
| | | | | $\Theta(n^2)$ | | | | |
| | | | $\Theta(n^2)$ | | | | | |

Problem			Algorithm			Implementation		
Best	Avg	Worst	Best	Avg	Worst	Best	Avg	Worst
$\mathcal{O}(n^4)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\Theta(n^2)$			$\Omega(n^3)$
$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$		$\mathcal{O}(n^3)$	$\Theta(n^2)$	$\Omega(n^3)$		$\Omega(n^3)$	$\Omega(n^3)$
$\mathcal{O}(n^3)$			$\Theta(n^2)$	$\Omega(n^3)$	$\Omega(n^3)$	$\Omega(n^3)$	$\Omega(n^3)$	$\Omega(n^3)$



- Solution:** 4500 seconds

- (b) A program requires 5 days to brute force attack a password of 48 bits. Since the running time is $\Theta(2^n)$ about how long would it take for the program to brute force attack a password of 256 bits?

Solution: 5×2^{208} days

- (c) If a program required 5 days to brute force attack a password 48 bits, how long would it take to attack a password of 256 bits if the running time were $O(n^2)$ instead of exponential?

Solution: 142 days

9. [9 pts] Give the tightest asymptotic average case upper and lower bounds you know for the following scenarios:

- (a) Deleting the 20th element of an array of size n when order doesn't matter

Solution: $\Theta(1)$

- (b) Deleting the 20th element of an array of size n keeping everything else in the same order?

Solution: $\Theta(n)$

The last one might be the 20th.

- (c) Finding the k smallest items in an unsorted array of size n

Solution: $\Theta(n)$

quick select, just for one time recursion.

- (d) Deleting an element from a heap of size n .

Solution: $\Theta(\log n)$ where n is no of elements in the heap.

- (e) The best algorithm finding the middle element (based on value) in a sorted array.

Solution: $\Theta(1)$

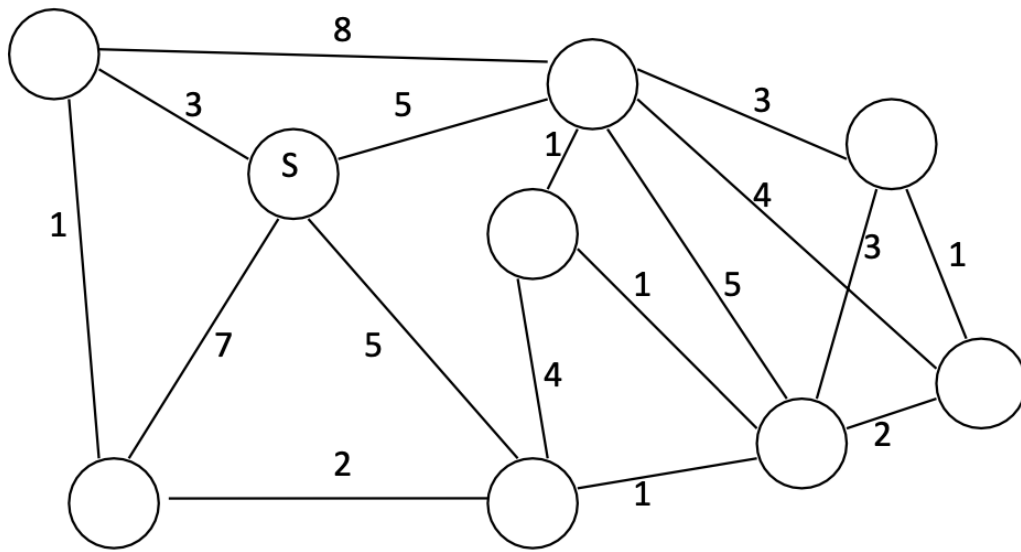
Just look it up middle index.

- (f) The best algorithm searching in a sorted linked list to determine if a specific element (based on value) is present.

Solution: **Solution:** $\Theta(n)$

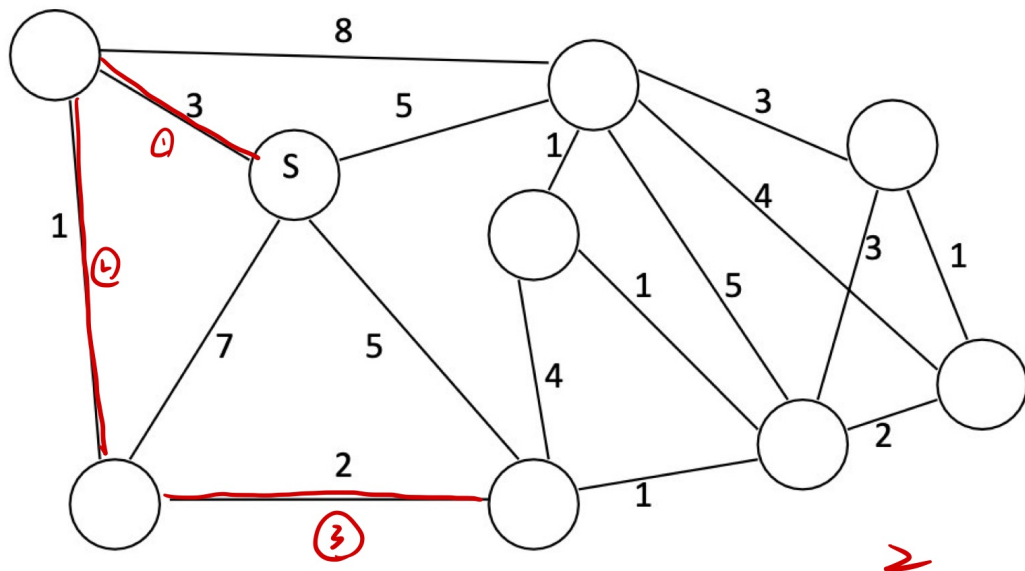
search all.

10. [10 pts] Consider the following graph. For any questions needing a starting vertex, use vertex S as the starting vertex.



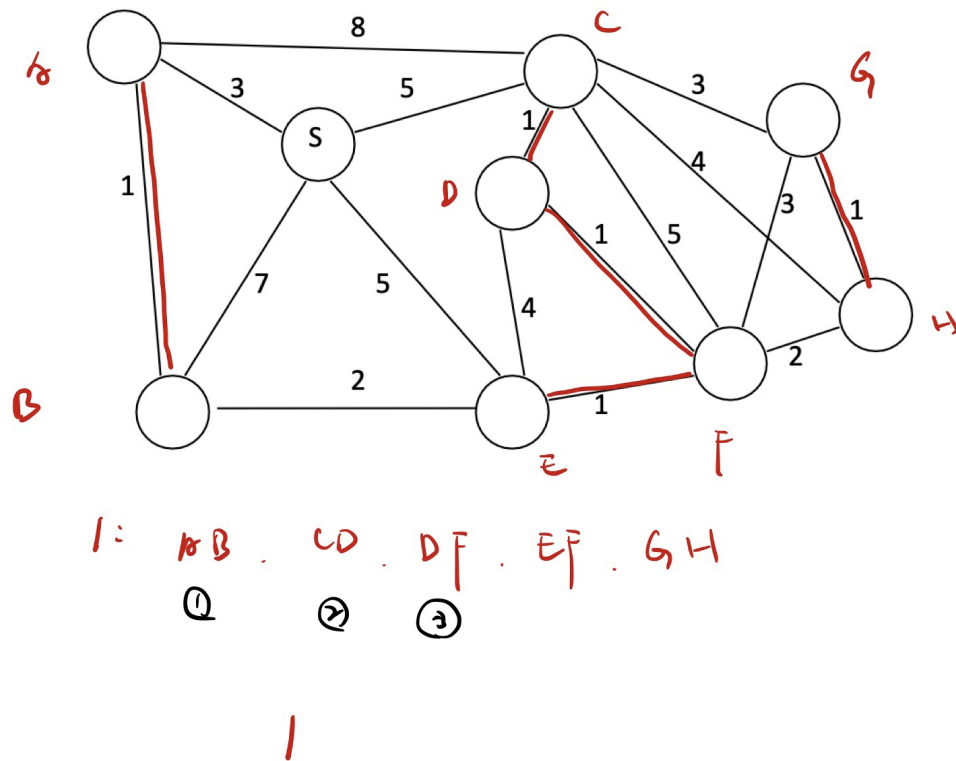
- (a) What is the value of the third edge chosen when finding a minimum spanning tree using Prim's algorithm?

Solution: 2



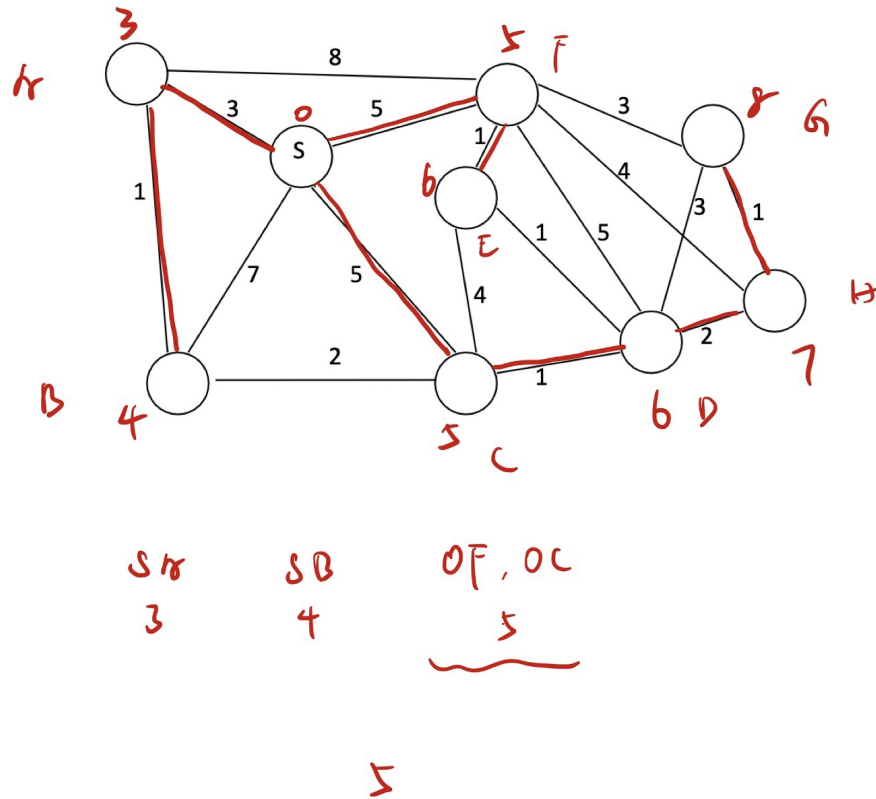
- (b) What is the value of the third edge chosen when finding a minimum spanning tree using Kruskal's algorithm?

Solution:



- (c) When using Dijkstra's algorithm to find the shortest path from S to all vertices, what is the value of the third edge chosen?

Solution:

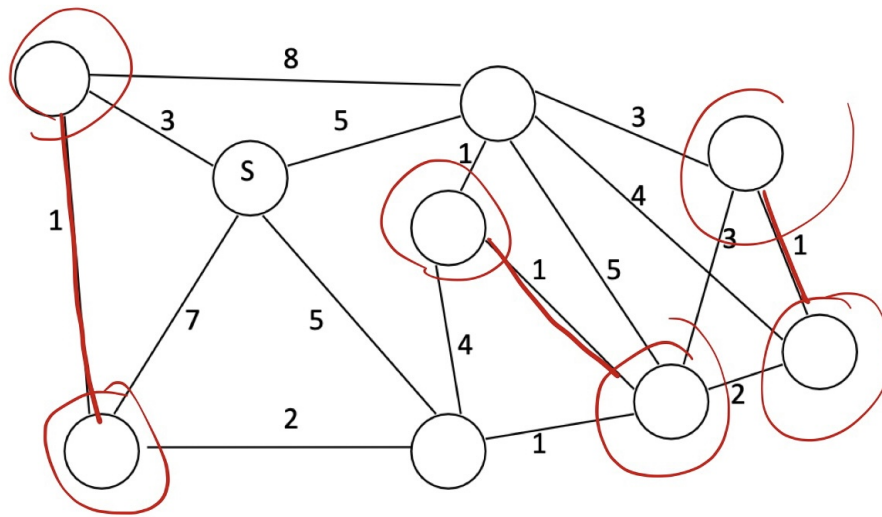


(d) How many components are in the graph?

Solution: 1

(e) What is the minimum number of edges you need to remove so the graph will have an Euler Tour? Mark the edges you would remove.

Solution: Euler Tour: visit each edge exactly once and end up of your start vertex.
3



Even degree

11. [6 pts] A particular problem requires 2 seconds to process 200 items and is $\Theta(n^3)$. You want to process 4000 items. You have a choice to either use a computer that is 10 times faster (allowing it to process 200 items in 0.2 seconds) or use the same computer with a different algorithm that still processes 200 items in 2 seconds, but has a growth rate that is $\Theta(n^2)$.

(a) Which is the faster choice for 4000 items?

Solution: Different

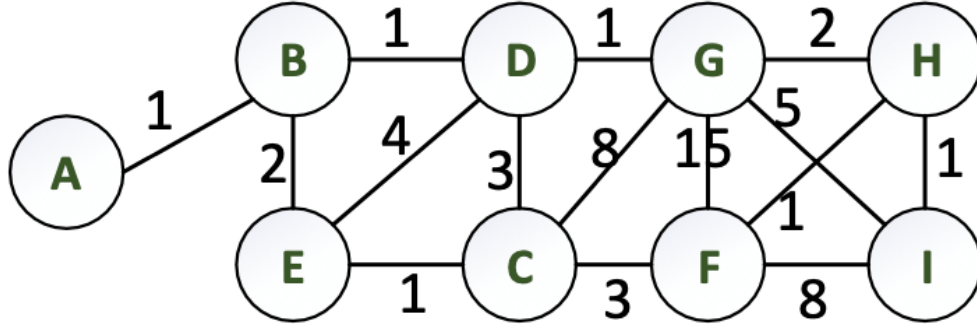
(b) For what input sizes is the faster computer better?

Solution: $n < 2000$ items

(c) For what input sizes is the $\Theta(n^2)$ algorithm better?

Solution: $n > 2000$ items

12. [9 pts] You live in city G. You want to know the cost to travel from city G to all other cities (A,B,C,D,E,FH and I). The edges of the graph below represent the cost to travel the roads between various cities. If an edge doesn't exist, then there is no road between those two cities. In this particular scenario, even though the roads have a different cost, it takes most of a day to travel each road. Therefore, you must spend the night at each intermediate city (vertex at an additional cost of 3.)

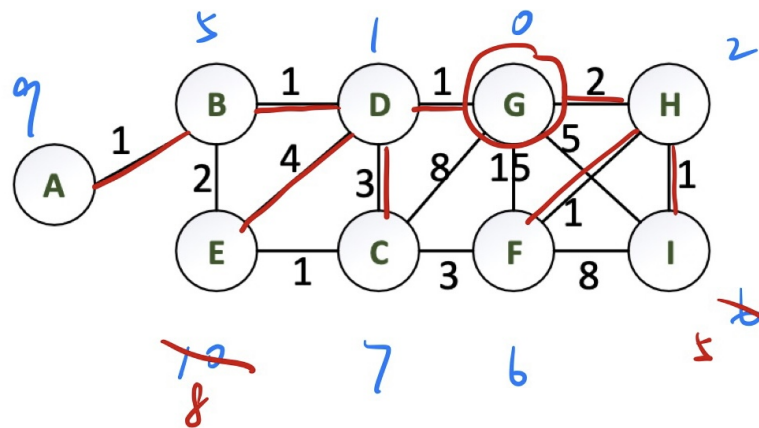
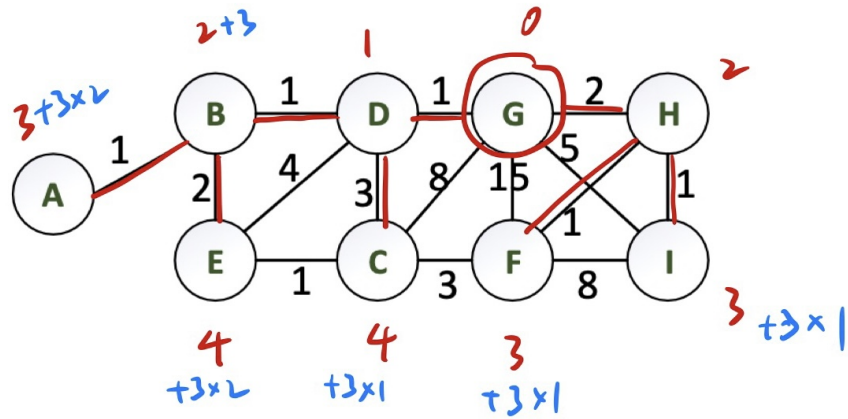
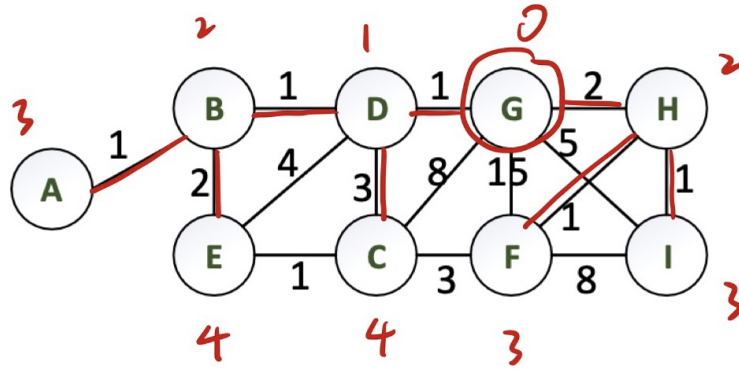


- (a) How would you modify Dijkstra's Single Source Shortest Path algorithm to find the cost from city G to all other cities in the graph with a vertex costing 3 to pass through it?

Solution: When passing through the city, the cost is 3 plus cost to reach the city instead of just the cost to reach the city.

- (b) What is the order you reach the cities in your adjusted algorithm.
 (c) Write the cost to reach each city from City G by its vertex in the graph.

Solution: (b) (c)



0 1 2 5 5 6 7 8 9
 G D H (B I) F C E 10
 (L B)