CS 5/7350 - Test#1
March 2, 2022

Name: Bingying Liang
ID: 48999397

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

(a) Algorithm

Solution: Informally, an algorithm is any well-defined computational procedure that takes some value, as input and produces some value, or set of values, as output in a finite amount of time. An algorithm is thus a sequence of computational steps that transform the input into the output. An algorithm can be viewed as a tool for solving a well-specified computational problem. The statement of the problem specifies in general terms the desired input/output relationship for problem instances, typically of arbitrarily large size. The algorithm describes a specific computational procedure for achieving that input/output relationship for all problem instances.
A step by step procedure for solving a problem in a finite amount of time.

(b) Big Theta

Solution: Θ-notation characterizes a tight bound on the asymptotic behavior of a function. It says that a function grows precisely at a certain rate, based – once again – on the highest-order term. Put another way, Θ-notation characterizes the rate of growth of the function to within a constant factor from above and to within a constant factor from below. These two constant factors need not be equal. If can show that a function is both $O(f(n))$ and $\Omega(f(n))$ for some function $f(n)$, then you have shown that the function is $\Theta(f(n))$. For example, since the function $7n^3 + 100n^2 - 20n + 6$ is both $O(n^3)$ and $\Omega(n^3)$, it is also $\Theta(n^3)$.
Big Theta is a tight upper bound and lower bound on a function.

(c) Set

Solution: A set is a collection of distinguishable items. For example, $S = \{1, 2, 3\}$.

(d) Implementation

Solution: An implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programming and deployment.
An implementation is a reduction of an algorithm to an executable format.

(e) Entropy

Solution: Entropy is defined as the randomness or measuring the disorder of the information being processed in Machine Learning. Further, in other words, we can say that entropy is the machine learning metric that measures the unpredictability or impurity in the system.

(f) Graph

Solution: A graph is a set of vertices $|V|$ and a relation on those vertices called edges $|E|$.

2. [8 pts] You have a complete graph with $|V|$ vertices where $|V|$ is ≥ 2 . Each edge in this graph has a capacity of 7. You pick one vertex as the Start Vertex, S, and another vertex as the Sink Vertex, T. Since the is a complete graph, you will get the same answer regardless of which two vertices you pick.

   (a) What is the length of the shortest path between Vertex S and Vertex T.

   Solution: $E(S, T)$
   Directly connected, so 1 edge of capcity 7

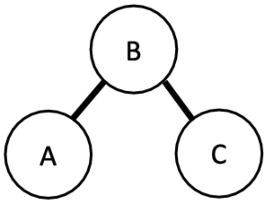
   (b) What is the maximum flow (in terms of $|V|$) between Vertex S and Vertex T.

   Solution: $7(|v| - 1)$

3. [6 pts] Argue that the problem, S, of sorting an unsorted array of integers of length greater than 100 elements is at least as hard - and maybe even harder - than the problem, L, of finding the five smallest values of the same array.

   Solution: Since a solver for problem S can be used to solve Problem L by sorting the array with S and getting the first 5 elements, Problem S must be just has hard or possibly harder than Problem L.

4. [6 pts] The following in-order traversal: A B C and the following pre-order traversal: B A C would form the following tree:



.

   Give a 3 vertex in-order traversal and a 3 vertex pre-order traversal that cannot form a tree. That is, no single tree can have both the in-order traversal and the pre-order traversal.

   Solution:
   in-order: ABC
   pre-order: CBA
   in-order: ABC
   pre-order: BCA

5. [6 pts] Using n_0 equal to 10, show that $f(n) = 7n^3 + 8n2 + 5n + 1$ is $\Omega(n2)$.

Solution:

$$\begin{aligned}
\Omega(g(n)) = \{f(n) :\ & \text{there exist positive constants } c \text{ and } n_0 \text{ such that} \\
& 0 \leq cg(n) \leq f(n) \text{ for all } n \geq n_0\} \\
& suppose\ g(n) = n^2 \\
& \therefore f(n) \geq cg(n) \\
& 7n^3 + 8n^2 + 5n + 1 \geq cn^2 \\
& 7n + 8 + \frac{5}{n} + \frac{1}{n^2} \geq c \\
& \lim_{n \to \infty} (7n + 8 + \frac{5}{n} + \frac{1}{n^2}) = \infty \\
& \because n_0 = 10 \\
& \therefore \text{exit positive constants } 1 \text{ and } n_0, \\
& let\ 7 \times 10 + 8 + \frac{5}{10} + \frac{1}{10^2} = 78.51 \geq 1 \\
& \therefore f(n) = 7n^3 + 8n^2 + 5n + 1 \text{ is } \Omega(n^2)
\end{aligned}$$

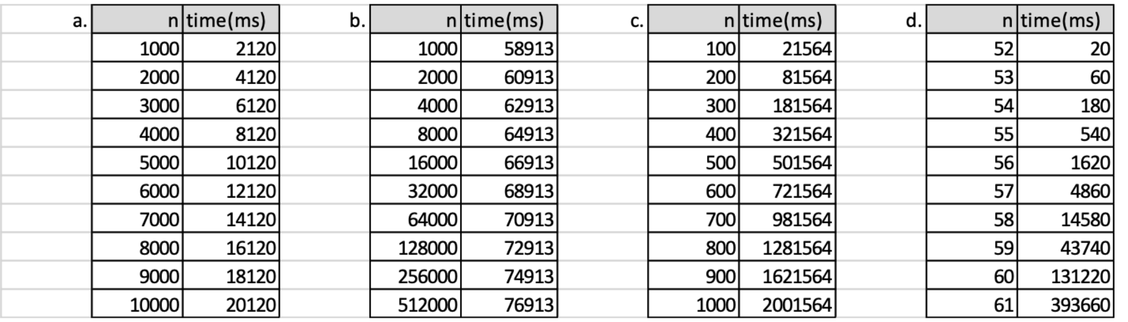
$$0 \leq c_1 n^2 \leq 7n^3 + 8n^2 + 5n + 1, \forall \geq 10$$

$$0 \leq c_1 \leq 7n + 8 + \frac{5}{n}, \forall \geq 10$$

$$c_1 \leq 70$$

Note that 70 is not a tight value, but works for the inequality.

6. [8 pts] You run different programs for various values of "n" and create 4 tables of the runtimes. Give the Asymptotic bounds that each of the tables support?



a.	n	time(ms)
	1000	2120
	2000	4120
	3000	6120
	4000	8120
	5000	10120
	6000	12120
	7000	14120
	8000	16120
	9000	18120
	10000	20120

b.	n	time(ms)
	1000	58913
	2000	60913
	4000	62913
	8000	64913
	16000	66913
	32000	68913
	64000	70913
	128000	72913
	256000	74913
	512000	76913

c.	n	time(ms)
	100	21564
	200	81564
	300	181564
	400	321564
	500	501564
	600	721564
	700	981564
	800	1281564
	900	1621564
	1000	2001564

d.	n	time(ms)
	52	20
	53	60
	54	180
	55	540
	56	1620
	57	4860
	58	14580
	59	43740
	60	131220
	61	393660

.

Solution:

(a) $\Theta(n)$

(b) $\Theta(\log_2(n))$

(c) $\Theta(n^2)$

(d) $\Theta(3^n)$

7. [6 pts] Answer the following Questions:

(a) Given that $M > 100$ and $(7^{311} \bmod M) = 1$; Find $7^{313} \bmod M =$

Solution: 49

$$7^{313} \bmod M = 7^{311+2} \bmod M = (7^{311} \bmod M) \times (7^2 \bmod M) = 49$$

(b) Compute 1/3 mod 9 =

Solution: Not exist.

(c) Compute $-\frac{1}{2} \bmod 11$

Solution: 5

8. [8 pts] A message contains the following number of each symbol:

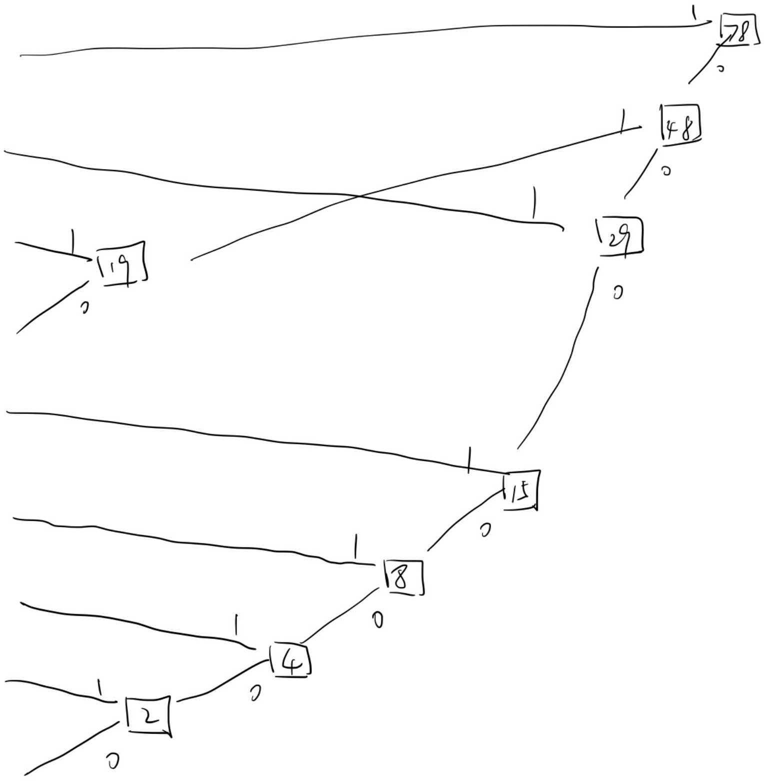
$$30A's, 14B's, 10C's, 9D's, 7E's, 4F's, 2G's, 1H\ and\ 1\ K.$$

Create a Huffman coding for each symbol:

How many bits are in the entire Huffman coded message?

Solution:

1	30 A's
1 0 0	14 B's
1 1 0	10 C's
0 1 0	9 D's
1 0 0 0	7 E's
1 0 0 0 0	4 F's
1 0 0 0 0 0	2 G's
1 0 0 0 0 0 0	1 H
0 0 0 0 0 0 0	1 K



$30 \times 1 + 14 \times 3 + 10 \times 3 + 9 \times 3 + 7 \times 4 + 4 \times 5 + 2 \times 6 + 1 \times 7 + 1 \times 7$

$= 30 + 42 + 30 + 27 + 28 + 20 + 12 + 14$

$= 72 + 57 + 48 + 26$

$= 129 + 74$

$= 203$ bits

203 bits

9. [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 performs the appropriate computations and sends the value 3 to person B. Person B chooses a random value of 4 and performs the appropriate computations:

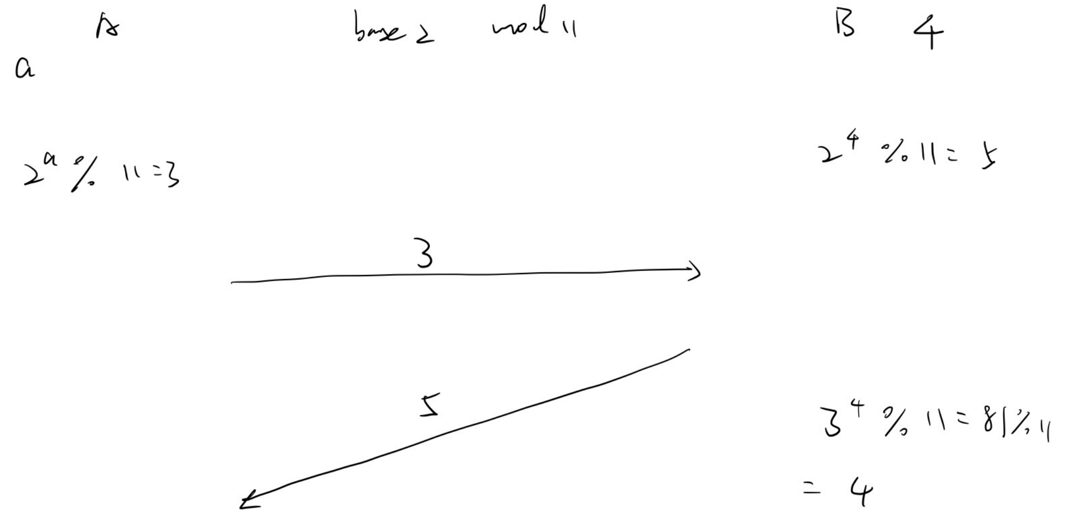
(a) What is the value Person B sends to Person A

Solution: 5

(b) What is the shared secret key between Person A and Person B

Solution: 4

Solution:



share secret key = 4

10. [9 pts] Consider two different algorithms that each solve a different problem.

- Implementation X solves Problem P_x and Implementation X is $\Omega(n)$ and $O(n^2)$
- Implementation Y solves Problem P_y and Implementation Y is $\Theta(2^n)$

Determine if each of these "Yes it is true", "Maybe it is true but doesn't have to be", or "No it is not true"

(a) Problem P_x is harder than Problem P_y : M

(b) Problem P_y is harder than Problem P_x: M

(c) Implementation X is harder than Implementation Y: N

(d) Problem X is $\Omega(n)$: M

(e) Problem X is $\Omega(n^2)$: M

Solution: Problem x is $O()$ is between $O(n^2)$ and $\Omega(n)$

(f) Problem X is $O(n)$: M

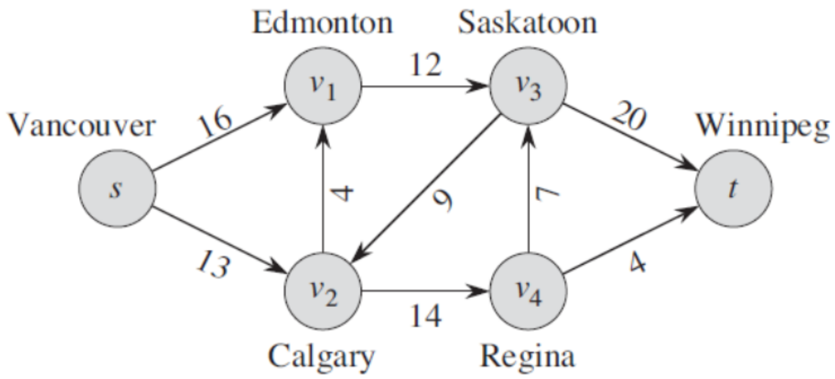
Solution: Problem x is $\Omega()$ is between $O(n^2)$ and $\Omega(n)$

(g) Problem X is $O(n^2)$: Y

(h) Implementation X is $\Omega(1)$: Y

(i) Implementation X is $O(2^n)$: Y

11. [8 pts] The graph below represents containers that are transported between these cities each day. You are determining the maximum flow from vertex S, Vancouver, to vertex T, Winnipeg, using the Ford-Fulkerson algorithm in the graph below.



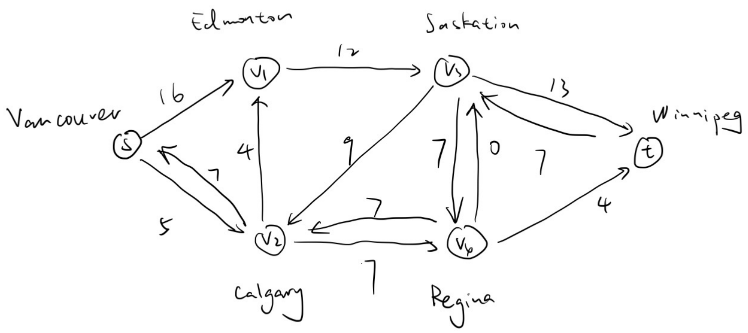
A Path Search finds the path $s \rightarrow v2 \rightarrow v4 \rightarrow v3 \rightarrow t$.

(a) How much flow is in this path?

Solution: 7

(b) What does the new graph look like after "removing" this flow

Solution:



A Path Search now finds the path $s \to v1 \to v3 \to v4 \to t$ on the new graph.

(c) Is it possible to find this path?

Solution: Yes

(d) If so, how much flow is in this path?

Solution: 4

12. [10 pts] Answer the following questions.:

(a) A program requires 6 seconds to process an input size of 40. If the running time is $\Theta(\sqrt(n))$ about how large of an input set could you process in 160 seconds?

Solution:

$$\frac{\sqrt{n}}{\sqrt{40}} \times 6 = 160 \to n \approx 28445$$

(b) A program requires 2 DAYS to brute force attack a password of 64 bits. Since the running time is $\Theta(2^n)$ about how MANY DAYS would it take for the program to brute force attack a password of 256 bits?

Solution:

$$\frac{2^{256}}{2^{64}} \times 2 = 2^{192} \times 2 = 2^{193} \; days$$

(c) A program requires 2 DAYS to brute force attack a password of 64 bits. About how MANY DAYS would it take for the program to brute force attack a password of 256 bits if the running time were $\Theta(n^2)$ instead of exponential?

Solution:

$$\frac{256^2}{64^2} \times 2 = 16 \times 2 = 32 \; days$$

(d) A program requires 4 milliseconds to process an input size of 1000. If the running time is $\Theta(n^4)$ about how many seconds would it take to process an input size of 1 million items?

Solution:

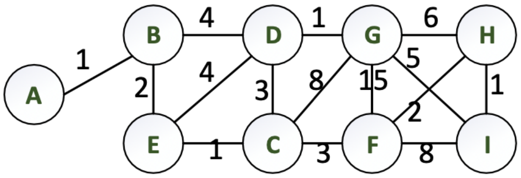
$$\frac{n^4}{1000^4} \times 4 \times 10^{-3} \; seconds = (\frac{10^6}{10^3})^4 \times 10^{-3} = 4 \times 10^9 \; seconds$$

(e) A program requires 4 milliseconds to process an input size of 1000. If the running time is $\Theta(n)$ about how long many seconds will it take to process an input size of 1 million items?

Solution: Solution:

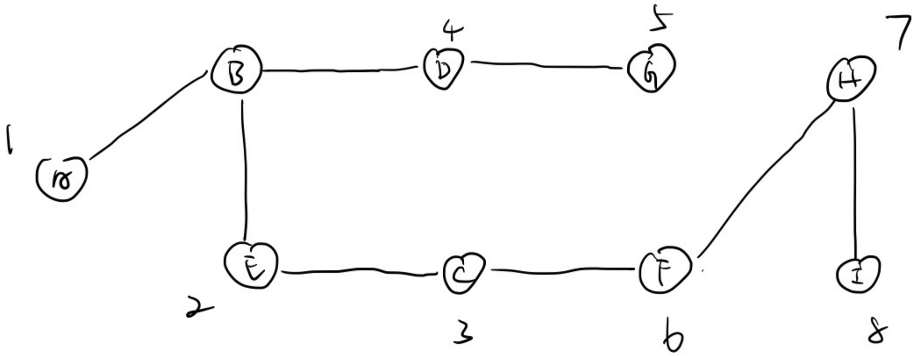
$$\frac{n}{1000} \times 4 \times 10^{-3}\ seconds = (\frac{10^6}{10^3}) \times 4 \times 10^{-3} = 4\ seconds$$

13. [5 pts] You live in city B. You want to know the cost to travel from city B to all other cities (A,B,C,D,E,F,H 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.



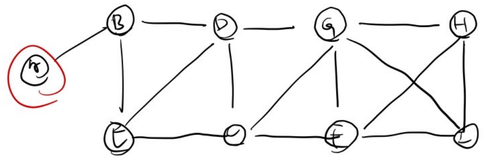
What is the order in which you explore the cities using Dijkstra's Single Source Shortest Path algorithm to find the cost from city B to all other cities in the graph?

Solution:

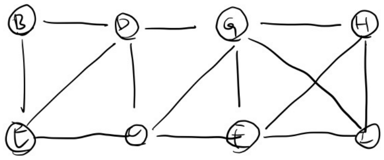


BAECDGFHI

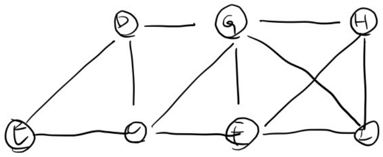
14. [5 pts] Give a smallest last vertex ordering for the graph in Problem #13. Circle in your ordering the first vertex you wrote down for that ordering.



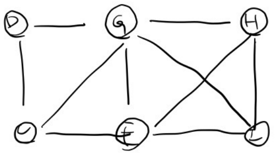
①



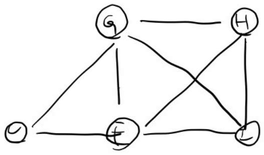
②



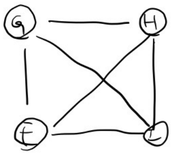
③



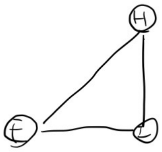
④



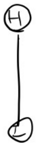
⑤



⑥



⑦



⑧

⑨

0	1	2	3	2	2	2	2	1
I	H	F	G	C	D	E	B	A