Computation Theory Notes:

Chapter 0

Question 0:

If a programmer understands the limitations and capabilities of a computer, they can more efficiently work to solve their problems.

Question 1:

The set *A* consists of all numbers between (and including) 1 and 30 that are divisible by 3.

Enumerating the element:

$A = \{e_1, \ldots, e_n\}$

Property of the element

Question 2:

A U B = {1, 2, 3, 4, 5, 6, 10}

A ∩ B = {10}

A \ B is the same as A minus the elements of B

Sets are unordered, but sequences have a specified order.
Sets are defined with parentheses: (7, 24, 38, 299)

Power Sets are a set of the original set and all of its subsets

Question 3:

***P*{6, 7, 8} = { {∅,} {6} {7} {8} {6, 7} {7, 8} {6, 8} {6, 7, 8} }**

Question 5:

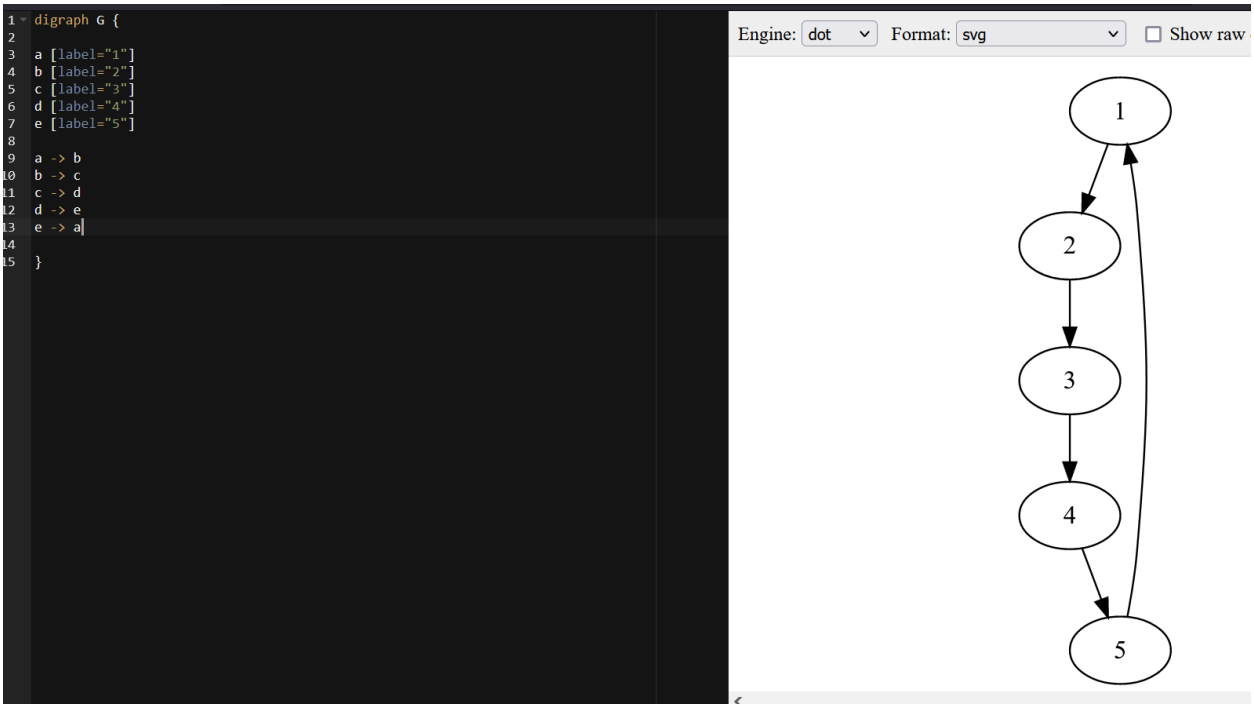
A^3 = All binary numbers from 0 to 7

Functions take an input and produce an output

Domain is the value of inputs

Range is the value of outputs

Question 6:



A graph is a tree if it is connected and has no simple cycles

Leaves are nodes with no children / they are the highest degree

Question 7:

We did it on the other assignment

Directed path: path where all arrows point in the same direction.

Alphabet: Non-empty, finite set

Symbols: Members of the alphabet

We use Greek letters to designate alphabets

Strings of characters are fundamental building blocks in CS

String(over an alphabet): finite sequences of symbols from that alphabet

Language: Set of "acceptable" strings

Assume w = 01001. |w| = 5

|...| is asking for the size of the set inside

If two strings are connected via o, like w o b, it's a concatenation

Concatenation: combine the strings in order

Question 8:

(ε), (0), (1), (00), (01), (10), (11), (000)

Question 9:

(P^Q) v (P^R)

(P v Q) ^ (P v R)

Question 10:

X cannot represent a "largest" number, as it would allow for X + 1 to exist and be larger.

If m and n have no common divisors, one must be odd.

Question 11:

2^0 one node / one leaf

Question 12:

n=1

1 = 1+1/2