
Number Theory.

1. *Factorize a number into a product of prime powers.*

Express 1001000 in the form $p_1^{k_1} p_2^{k_2} \cdots p_m^{k_m}$, where all p_i are different primes and k_i are non-negative integers.

2. *Divide numbers with remainders as in $n = qd + r$, express decimal digits.*

Assume that you have all 4 arithmetic operations and also two more operations: **div** (integer division) and **mod** (remainder).

For example $17 \text{ div } 3 = 5$, $(-17) \text{ div } 3 = -6$, $17 \text{ mod } 3 = 2$, $(-17) \text{ div } 3 = 1$.

Note. In Python **div** and **mod** are written as $17//3$ and $17\%3$ respectively.

Let n be a positive integer.

- (a) Express n as $n = 7q + r$, where $r \in \{0, 1, 2, 3, 4, 5, 6\}$, but q is any integer. Write formulas that express q and r for the given n .
 - (b) Express the last binary digit of n (it should be 0 for even n and 1 for odd n).
 - (c) Write an expression $d_4(n)$ for the 4rd decimal digit (from the right). It should return 0, if the number n has fewer than four decimal digits.
For example, $d_4(2345) = 2$ and $d_4(999) = 0$.
 - (d) Write an expression $h_3(n)$ for the 3rd hexadecimal digit (from the right). And it should be 0, if number does not have 3 hexadecimal digits.
For example $h_3(37FF_{16}) = 7$; $h_3(7FFF_{16}) = 15$, and $h_3(FF_{16}) = 0$.
3. *Find members of arithmetic progressions, also by modulo m .*
- Let (a_n) be an arithmetic progression defined by a recurrence: $a_0 = 3$ and $a_{n+1} = a_n + 7$.
- (a) How many members of this progression are 3-digit numbers (i.e. numbers between 100 and 999)?
 - (b) What is the last digit of the member a_{2021} ?
4. *Given two integers, find their GCD (also LCM) by Euclidean algorithm.*

- (a) Write the Euclidean algorithm execution to find $\gcd(1396, 482)$.
- (b) Express also $\text{lcm}(1396, 482)$ – the least common multiplier of both numbers.
- (c) The prime factorization for both numbers 7007 and 9100 are given:

$$\begin{cases} 7007 = 7^2 \cdot 11^1 \cdot 13^1, \\ 9100 = 2^2 \cdot 5^2 \cdot 7^1 \cdot 13^1. \end{cases}$$

Express the prime factorizations of the $\gcd(7007, 9100)$ and also $\text{lcm}(7007, 9100)$.

5. *Given integers, solve Bezout identity with Blankenship algorithm.*

Find a positive integer $d = \gcd(106, 16)$ and find some two integers x, y that solve the Bezout identity:

$$106x + 16y = d.$$

6. Given m and x , compute multiplicative inverse \bar{x} modulo m ; solve linear congruences.

Assume that you need to solve several linear congruences $3x \equiv a$ modulo 37 for various values of a .

$$\begin{cases} 3x \equiv 5 \pmod{37} \\ 3x \equiv 6 \pmod{37} \\ 3x \equiv 7 \pmod{37} \\ 3x \equiv 8 \pmod{37} \end{cases} \quad (1)$$

There is no division in modular arithmetic, but you can find inverse of 3 modulo 37 which is helpful to solve these congruences.

- (a) Find the multiplicative inverse of 3 modulo 37.
 - (b) Solve any two congruences from (1).
7. Convert periodic decimal numbers to rational fractions.
- (a) Convert the periodic fraction $0.2021(235) = 0.2021235235 \dots$ into a rational fraction m/n where $m, n \in \mathbf{Z}$.
 - (b) Express the number $0.0000(235) = 0.0000235235 \dots$ as the sum of infinite geometric series.
8. Given a decimal integer, convert it to binary, hexadecimal (and vice versa).
Convert 1000_{10} into binary and hexadecimal notation.

Graphs.

1. Count the number of graphs with a given property or parameter.
Count the number of graphs having exactly 4 vertices and exactly 3 edges. (All graph vertices are considered unique; graphs that differ by isomorphism are considered different graphs.)
2. Justify whether or not a graph has a particular subgraph, cycle or path.
Consider the full bipartite graph $K_{3,3}$ (two sets of 3 vertices; all 9 edges that connect vertices between different sets belong to this graph).
 - (a) Does it contain C_4 as a subgraph (a cycle with 4 vertices)?
 - (b) Does it contain C_5 as a subgraph (a cycle with 5 vertices)?
3. Convert between graph representations. Check if a graph is bipartite, complete, or connected.
A graph $G(V, E)$ has 5 vertices labeled with numbers: $V = \{0, 1, 2, 3, 4\}$. It has the following adjacency list representation:

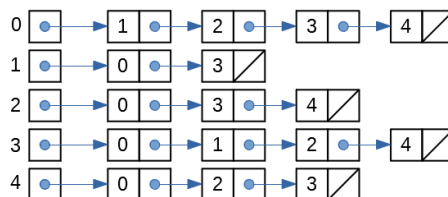


Figure 1: Adjacency lists for a graph.

- (a) Create the diagram for this graph showing all the vertices and edges.

- (b) Create the matrix representation for this graph.
4. *Given a tree, check the condition for a Euler circuit (or path) and find it.*
Figure 2 shows an undirected graph.

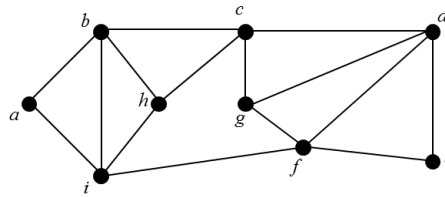


Figure 2: Graph to find Euler circuits or paths.

- (a) Does the graph have a Eulerian circuit? (Show the circuit or prove that it does not exist.)
- (b) Does the graph have a Eulerian path? (Show the path or prove that it does not exist.)
5. *Given two graphs prove or disprove they are isomorphic.*
For the graph pairs shown in Figures 3 and 4 either show that both graphs in a pair are isomorphic, or show that they cannot be isomorphic.

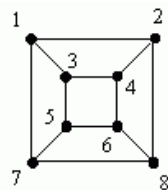
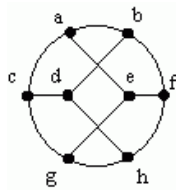


Figure 3: Graphs to test isomorphism (1)

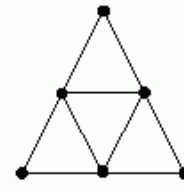


Figure 4: Graphs to test isomorphism (2)

Trees

1. *Given the count of vertices, edges, height or other parameter, estimate other parameters.*
There is a rooted tree T with 100 vertices; any inner node has no more than 2 vertices.
- (a) Find the smallest possible number of leaves that can exist in T . Find the largest number of leaves that can exist in T .
- (b) Find the smallest possible height of T .
2. *Given an n -ary tree and some parameters, estimate other parameters.*
Let T be a full 10-ary tree with exactly 2021 vertices.
- (a) Find the number of inner nodes in T .
- (b) Find the number of leaves in T .
- (c) Find the number of edges in T .
- (d) What can be the length of the longest path in T (create T so as to make this path as long as possible).

3. *Convert between representations: tree diagrams, lists of edges, or traversals.*

The post-order traversal of some rooted, ordered tree T is as follows:

$$v_4, v_{13}, v_9, v_{10}, v_5, v_6, v_1, v_2, v_{11}, v_{14}, v_{15}, v_{12}, v_7, v_8, v_3, v_0.$$

In this tree v_0, v_1 have tree children each; v_3, v_5, v_7, v_{12} have two children each; v_9 has one child.

- Represent this tree as a diagram (root vertex at the very top; parents connected with their children).
 - Represent this tree as adjacency lists (each edge in this tree is directed from a parent to a child).
4. *Given a prefix, infix or postfix notation, convert it into the syntax tree or other notation(s).*
Consider the following predicate-logic expression:

$$\neg \left(\forall a \in H \exists b \in H \exists c \in H \left(R(a) \vee (K(a, b) \wedge K(a, c) \wedge b \neq c \wedge R(b) \wedge R(c)) \right) \right).$$

Boolean operations \vee and \wedge in this formula are *binary* (they are nodes which need two children each), boolean operation \neg and all the quantifiers ($\forall a \in H, \exists b \in H$, etc.) are *unary* (they are nodes which need just one child), but the predicates such as $R(a), K(a, b)$ are leaves.

- Draw the syntax tree for this predicate logic expression. Assume that disjunction and conjunction are both left-associative.
 - Visit all the vertices of this syntax tree in the pre-order sequence.
5. *Given a directed or an undirected graph, do a DFS and BFS traversal.*

Figures 5 and 6 show directed and undirected graphs respectively.

Create their BFS and DFS trees (their roots are in vertices A and 1 respectively). Visit all children in alphabetical order (or the order of increasing numbers). If at some point you run out of vertices, pick the alphabetically smallest vertex among the remaining ones and start the process. You should create 4 different trees (BFS for the directed graph; BFS for the undirected graph; DFS for the directed graph; DFS for the undirected graph).

Note. You do not need to show forward edges, back edges, cross edges in your pictures; just the tree edges are enough.

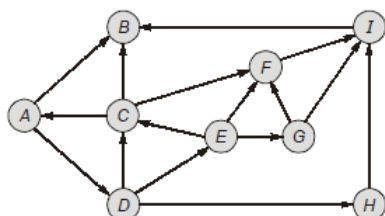


Figure 5: Directed graph for DFS/BFS

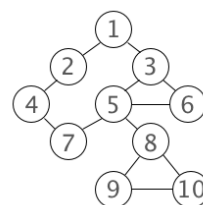


Figure 6: Undirected graph for DFS/BFS