The purpose of this tutorial is to familiarise you with some fundamental concepts from discrete mathematics that are useful for designing and analysing data structures and algorithms. For a more complete account, including more detailed explanations and more examples, study Sections 1.2.1, 1.2.2 and 1.2.3 in the recommended reading "Algorithm Design and Applications" by Goodrich and Tamassia. The topics covered there, and which you should revise, are:

1. Summation notation Σ, and formulas for the sum of certain numeric sequences, e.g., $\Sigma_{i=1}^{n} i$ and $\Sigma_{i=1}^{n} 2^i$, etc.
2. Logarithms and exponents
3. Floor and ceiling functions
4. Justification by counterexample
5. Justification by contrapositive
6. Justification by contradiction
7. Justification by induction
8. Basic probability

In addition to that, you should be familiar with basic set notation and basic logic. A set is a group of objects represented as a unit. The objects in a set are called its members or elements. One way to describe a set is to list its elements inside braces, e.g., $S = \{7, 21, 57\}$. The symbols $\in$ and $\notin$ denote set membership and nonmembership. So $7 \in S$ but $8 \notin S$.

Warm-up

Problem 1. Recall that $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$.

a) Use this formula to compute $\Sigma_{i=1}^{20} i$.

b) Write a formula for $\Sigma_{i=m}^{n} i$ and evaluate it for $m = 10, n = 20$.

c) Write a formula for $\Sigma_{i=n}^{2n} i$, and evaluate it for $n = 20$.

Problem 2. Examine the following formal descriptions of sets so that you understand which members they contain. Write a short informal English description of each set.

a) $\{1, 3, 5, 7, ...\}$

b) $\{..., -4, -2, 0, 2, 4, ...\}$

c) $\{n \mid n = 2m \text{ for some } m \in \mathbb{N}\}$

Problem 3. Write formal descriptions of the following sets.

a) The set containing the numbers 1, 10, and 100

b) The set containing all integers that are greater than 5

c) The set containing all natural numbers that are less than 5

d) The set containing nothing at all

Problem 4. Let A be the set $\{x, y, z\}$ and B be the set $\{x, y\}$.

a) Is A a subset of B?

b) Is B a subset of A?

c) What is $A \cup B$?

d) What is $A \cap B$?

e) What is $A \times B$?

f) What is the power set of B?

Problem 5. Given the array:

$$A = [3, 7, 2, 9, 1]$$

Perform the following operations in order:

1. Insert the number 5 at index 2.
2. Remove the last element from the array.
3. Sort the array in ascending order.
4. Reverse the sorted array.

What is the final resulting array?

Problem solving

Problem 6. A **leap year** is a year that is evenly divisible by 4, except for years that are evenly divisible by 100, unless they are also evenly divisible by 400.

Leap Year Rules:

- A year is a leap year if it is divisible by 4.
- If the year is also divisible by 100, it is **not** a leap year.
- However, if the year is divisible by 400, then it **is** a leap year.

Example Cases:

Year	Leap Year?	Reason
2024	Yes	Divisible by 4, not by 100
1900	No	Divisible by 100, but not by 400
2000	Yes	Divisible by 400
2023	No	Not divisible by 4

a) Design an algorithm to determine whether a given year is a leap year.

b) Analyze the time complexity and space complexity of your solution.

Problem 7. Write an algorithm to reverse an array in place (without using extra memory).

Example Input:

$$[1, 2, 3, 4, 5]$$

Expected Output:

$$[5, 4, 3, 2, 1]$$

a) Implement the algorithm in pseudo code.

b) What is the time complexity of your algorithm?

c) What is the space complexity, and why?

Problem 8. Given an array with n integer values, we would like to know if there are any duplicates in the array. Design an algorithm for this task and analyze its time complexity.