Computing Problems

Last Updated: August 17th, 2026

1 Computing Problem Definition

Informally, when we think of a problem, we think of a situation that needs to be resolved (i.e. solved). Moreover, in computer science we think of a computing problem as having the following properties.

Definition 1.1. A **computing problem** is a collection of situations that share a common theme, and each of which requires a solution.

- Each situation is referred to as a **problem instance**, and represents a concrete example of the general problem.
- Each problem instance can be represented by a unique word over some alphabet, meaning that no two instances map to the same word. The word may then be encoded into a binary word so that the problem instance can be stored in computer memory.

1.1 Three types of computational problems in relation to this course

Decision The solution to a problem instance is either Yes or No, equivalently True (1) or False (0). An instance x for which the solution is 1 (respectively, 0) is called a **positive instance** (respectively, **negative instance**).

Optimization The solution to a problem instance x is a number that represents the greatest (or least) quantity of some entity that is associated with x

Miscellaneous Any computation problem that is neither a decision nor an optimization problem

Example 1.2. For each of the following problems, determine if it is a decision, optimization, or miscellaneious problem.

a. An instance of the `Prime` problem is a natural number $n \geq 0$, and the problem is to decide if n is a prime number, meaning that its only natural divisors are 1 and n.

b. An instance of the `Maximum Subsequence Sum (MSS)` is a sequence (array) a of integers. The problem is to determine the greatest sum that can be made by any subsequence of a. For example, determine the maximum subsequence sum for any subsequence of

$$3, -4, 2, 5, -2, -4, 0, 2, 3, -2, 1.$$

c. An instance of `Sort` is an array a of integers. The problem is to produce another array b whose members are the members of a, but in sorted order.

d. An instance of `Fallible` is a Boolean formula F. Is there an assignment that can be made to the variables of F so that F evaluates to 0? For example, given the logical formula $F(x, y) = x \to (y \to x)$, can x and y be assigned Boolean values that force F to evaluate to 0?

2 Problems as Sets

Notice that every problem is given a name consisting of one or more words along with an associated acronym if necessary. We often write a problem's name to represent the *set* of all problem instances. For example,

$$f : \texttt{Sort} \rightarrow \{0, 1\}$$

is a function whose inputs are instances of problem `Sort`, and whose outputs are 0 or 1. For example, we may define $f(a) = 1$ iff integer array a is a sorted array, and $f(a) = 0$ otherwise.

3 The Size of a Problem Instance

Definition 3.1. Given a computing problem A and an instance x of A, $|x|$ denotes the **size** of x and equals the number of bits needed to binary-encode x. The notation $|x|$ is often useful when speaking abstractly about a generic problem, and an instance x of that problem.

Definition 3.2. Given a problem A, a **size parameter** for A is a parameter that may be used to (approximately) represent the size of an instance of A. Given an algorithm $\mathcal{A}$ that decides A, its size parameters allow one to describe the number of steps (and/or amount of memory) required by $\mathcal{A}$ as a function of the size parameters.

Example 3.3. Provide reasonable size parameters for each problem described in Example 1.2.

Solution.