Designing an algorithm may be a nontrivial task

Lecture 1

An algorithm is a method to solve a problem. The problem need not be just a computational problem. An algorithm must satisfy the following properties. Please convince yourself about their importance.

- Finite description:
  It must be possible to describe the algorithm as a finite sequence of steps.
- Finite Execution:
  For any input instance of the problem, it should take finite number of steps to compute and output the solution. In particular, the number of steps executed by the algorithm should be bounded by some function of the input.
- Correctness:
  For each *possible* instance of the problem, the algorithm **must** output a correct solution.

You have already done a course on data structures and algorithms. Based on the problems you discussed/solve there, you might have felt that it is quite easy to design an algorithm but it is usually challenging to design an efficient algorithm. Through an interesting problem, I shall convince you that even the design of an algorithm is quite challenging sometimes. As a warm up, we shall first consider an easy problem.

Stationary Friend

You are standing at the crossing of two perpendicular roads leading to infinity. Your friend is standing at some point on one of these roads. See Figure 1.

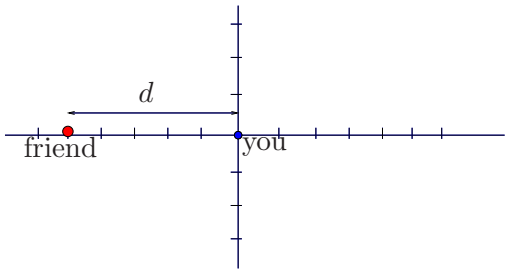

Figure 1: Searching for your stationary friend

Note that

- You don't know the distance to your friend.
- You don't know the direction (road) leading to your friend.

You may walk along the roads. Your aim is to reach your friend. Design an algorithm for this problem.

Exercise 1:
Design an algorithm to search for your friend such that the distance traveled is $O(d^2)$. Make sure this algorithm has the three key properties of an algorithm.
Exercise 2:
Design an algorithm to search for your friend such that the distance traveled is $O(d)$.

Fully internalize these algorithms before proceeding to the main problem on the following page.

1 Mobile friend

Your friend is moving along a long straight road at a speed of 1 km/min. In the beginning he/she is at distance d km from the origin, where d is some integer. See Figure 2.

- You don't know the value of d.
- You don't know the direction along which your friend is moving.

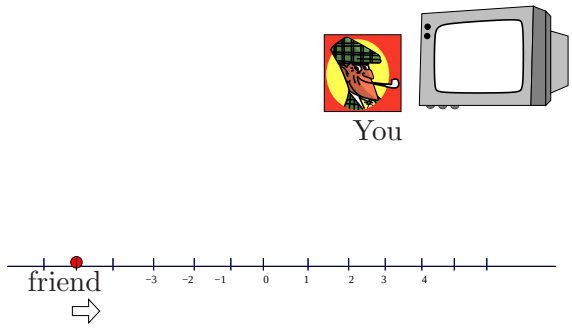

Figure 2: How to search for your mobile friend ?

There are video cameras installed at each kilometer stone along the road. You are sitting in a room with a monitor in front of you. Every minute, you can input an integer i, and you are able to watch the scene at the ith kilometer stone. So if your friend is there at that moment, he gets located by you and your goal is achieved.

Exercise 3:
Design an algorithm to locate your mobile friend. The algorithm should ensure that you locate your friend in $O(d)$ minutes.

Extend the algorithm suitably to the following apparently even harder problem.
Exercise 4:
Your mobile friend is moving in a 3-D grid. Moreover he/she is allowed to change his direction at most 100 times. Of course, he/she does not inform you while changing direction.

It is quite natural to feel that there won't be any algorithm for the problem mentioned above. (Similar is a reaction whenever one starts working on a research problem). If you are interested in this problem and have enough perseverance to keep your interest alive for next 4 months, it is most likely that you will solve it yourself.

The impact of efficient algorithms in real life

Every year new and more efficient algorithms are designed for nearly 300 problems. However, it is also a fact that hardly 1% of them are eventually implemented. But there are many simple but importance problems even in real life that demand efficient algorithm. Consider the following problem.

You have a PC which has 64MB RAM and a hard disk with capacity of a few Tera Bytes (10^{12} bytes). The hard disk stores one trillion numbers. Your aim is to find the median of these numbers. For this you may scan the hard disk. Your aim is to minimize the number of scans.

Would you believe that there is an algorithm that will make only 2 scans of the hard disk to compute the median ? Yes, indeed. We shall discuss this algorithm sometimes in this course.

With these nice problems/puzzles, we begin this course. After all, the zeal to solve a puzzle or computer problem is (or should be) the main driving force for me and you to be here in this course.

Wishing you true joy pondering over the above and many more problems that await you in the course!