Module 37: Randomized Algorithms

This module 37 focuses on Randomized algorithms. The module also introduces basics of randomized algorithms, random numbers and introduces some elementary problems. The objectives of this module are

- To understand the Randomized algorithms
- To understand Pseudorandom numbers and its generation
- To understand basic Randomized algorithms

NP-Hard Problems

Many real world problems are intractable problems. An intractable problem is a problem that requires more computer resources such as space and time. But, these problems cannot be ignored as one encounters these problems in daily life. For example, problems like scheduling or weather forecasting are hard problems. Hence, there is a necessity to explore alternative ways to solve these problems. One such approach for attempting to solve difficult problems is randomized algorithms. Another approach is approximation algorithms. Let us discuss about randomized algorithms now.

Randomized Algorithms

What is randomness? Randomness is a state of the system whose behaviour follows no deterministic or predictable pattern. Some of the daily encounters like gambling, puzzles, decision making pprocess and heuristics are examples of randomness.

Randomness is used as a computing tool by randomized algorithms for algorithm design. Randomized algorithms are also called probabilistic algorithms.

It can be recollected from module 1 that an algorithm takes an input, process it and generates an output. This is shown in Fig. 1.

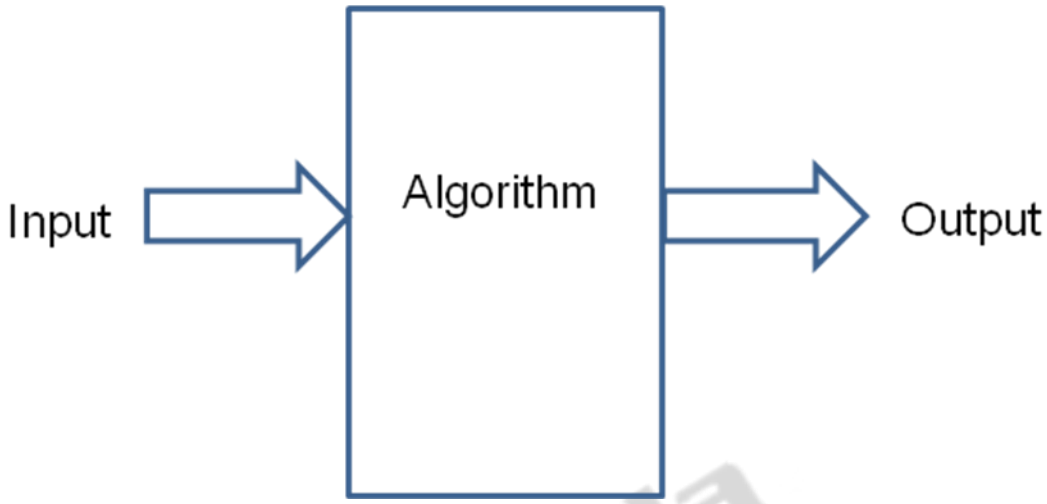

Fig. 1: Algorithm Environment

Algorithms can be classified into deterministic algorithms and randomized algorithms. The output is always fixed for deterministic algorithms.

Randomized algorithms are on the other hand [1,2,3] is as shown in Fig. 2.

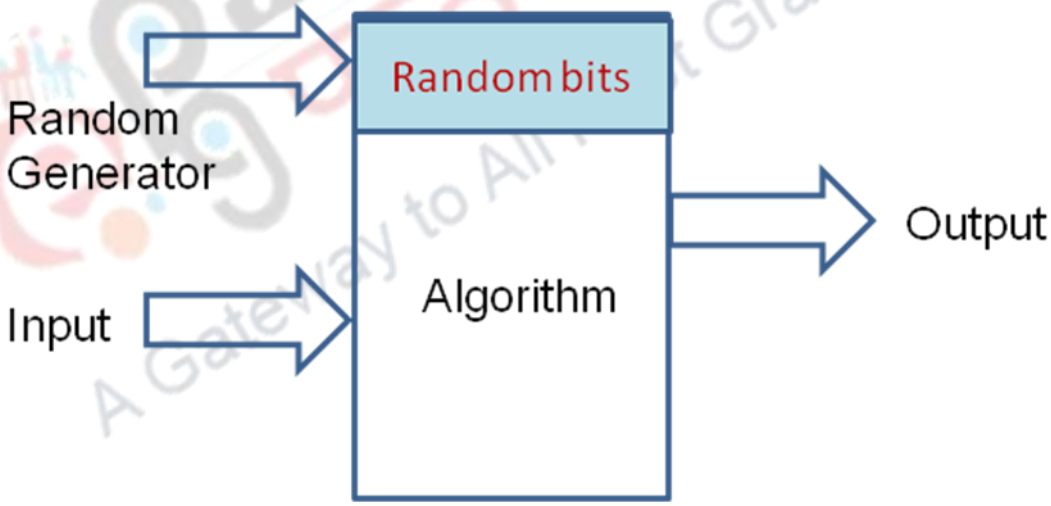

Fig. 2: A Randomized algorithm

It should be noted that randomized algorithms output is based on random decisions and its output is based on probability. There would be negligible errors on the long run.

In short, Randomized Algorithms are dependent on inputs and use random choices as part of the logic itself.

What are the advantages of randomized algorithms? Some of the advantages of randomized algorithms are given below:

- Known for its simplicity
- very Efficient
- Computational complexity is better than deterministic algorithms

Some of the disadvantages of randomized algorithms are as given below:

- Reliability is an issue
- Quality is dependent on quality of random number generator used as part of the algorithm

But randomized algorithms are very popular and are useful to solve many problems in computer science domain effectively. Let us discuss some of the design principles that are useful for randomized algorithm design.

Concept of witness

This is one of the important design principles for randomized algorithms. The concept of witness is about checking whether given input X has property Y or not. The core idea is the concept of witness that gives a guarantee. Some of the problems like random trials and Primality testing can be solved using this concept of witness.

Fingerprinting

Fingerprinting is the concept of using a shorter message representative of a larger object. This representative is called fingerprinting. If two large strings need to be checked, then instead of comparing two larger strings, two fingerprints can be compared. The problem of comparing larger strings can be done using this design principle.

Randomized Sampling and Ordering

Some problems can be solved by random sampling and ordering. This is done by randomizing the input distribution or order or by partitioning or sampling randomly. Some of the problems that use this principle are hiring problem and randomized quicksort.

Foiling adversary

This is another useful principle. This can be viewed as a game between a person and an adversary with both are attempting to maximize their gains. This can be view as a selection of algorithm from a large set of algorithms

Types of Randomized Algorithms

There are two types of algorithms. One is called Las Vegas Algorithms and another is called Monte Carlo Algorithms.

Las Vegas Algorithms have the following characteristics

- always correct
- "probably fast"

Randomized quicksort is an example of Las Vegas algorithm. It is faster than the traditional quicksort algorithm and its results are always correct.

Monte Carlo algorithms were designed by Nicholas Metropolis in 1949. Unlike Las Vegas algorithms, Monte Carlo algorithms give results that are mostly or probably correct. These algorithms have guaranteed running time unlike Las Vegas algorithms. Primality testing problem can be solved using Monte Carlo algorithms.

Complexity Class

Like P and NP classes for deterministic algorithms, randomized algorithms also can be grouped together as class of problems. Some of the classes are given below:

RP Class

RP class is a set of decision problems solvable with one-sided error in polynomial time. It is an abbreviation of Random Polynomial algorithms. What is a one sided error? If the correct answer is 'NO', then the algorithm always returns 'NO' as the answer. But, if the ccorrect answer is 'YES', return algorithm result is associated with a probability. In other words, the algorithm output would be 'YES' with probability $\geq$ ½.

Monte Carlo algorithms belong to class RP.

ZPP Class

ZPP is a class of decision problems that are solvable in expected polynomial time. Las Vegas algorithms are examples of ZPP class.

There is a theorem that defines the hierarchies as follows:

$$P \subseteq ZPP \subseteq RP \subseteq NP$$

Random Numbers

One of the primary requirements of good quality randomized algorithms is the quality of its random number generator. The quality of the random number generator determines the reliability of the randomized algorithm. Let us discuss about them now.

A random number generator generates a random number. The true random numbers are based on radioactive decay; flip of coins, shot noise, radiations. One of the characteristic of the "true" random number generator is that the generated number should not appear again. But, based on the memory and processor limitations, generation of such number is often difficult. So, Pseudo random numbers are generated. Pseudo-random numbers are as good as true random numbers in most of the situations.

Pseudo-random numbers are generated using software applications and can be recreated if formula is known. But, Pseudo-random numbers are sufficient for most of the purposes.

Some of the characteristics of "Good" random numbers are

- Efficiency
- Deterministic Property
- Uniformity
- Independence
- Long cycle

There are many algorithms are available for generating pseudorandom numbers. One simplest algorithm is called Linear Congruential Generator (LCG). The formula of LCG is given below:

- X_{i+1} = (a * + b) **%** m;

Here, a and b are large prime numbers, and m is 2^{32} or 2^{64}. The initial value of X_i is called a seed.

Often a permutation array is created by generating random numbers and storing it as an array. The array of random numbers is called permutation array. The steps of creating a permutation array are given as follows:

1. Let index = 1
2. Generate random number
3. Swap A[index] and random number
4. Fill the array

The formal algorithm based on [1,2,3] is given as follows:

```
Algorithm random-array(A)
Begin
  for index = 1 to N
  k = random();
  Exchange A[index] and A[k]
  End for
End
```

Hiring problem

Hiring problem is a problem of hiring a secretary among a group of secretaries. This problem can be solved using deterministic and randomized algorithm.

Informally, the steps of hiring problem [1,3] is given as follows:

1. Initial candidate is best candidate
2. Interview new candidate
3. If new candidate is better, then hire new candidate and old candidate is fired.

What is the complexity analysis? Conducting interview and hiring costs something. If n candidates interviewed and m candidates hired. In that case, the total cost of the algorithm would be $O(m \times C_{hire} + n \times C_{\text{Interview}})$.

The computational complexity can be improved by randomized hiring algorithm. The improvement comes because of shuffling the input. By random order of the input, the algorithm becomes randomized algorithm. The steps of the randomized hiring problem are given as follows:

1. Randomly permute array A
2. Let Initial candidate is best candidate
3. Interview new candidate
4. If new candidate is better, then hire new candidate and old candidate is fired.

Randomized analysis of this algorithm is discussed in the next module.

Summary

One can conclude from this module 37 that

- Randomized algorithms are effective
- Pseudo-random numbers are useful in randomized algorithms.
- Hiring problem can be solved using randomized algorithms.

References:

1. *S.Sridhar , Design and Analysis of Algorithms , Oxford University Press, 2014.*
2. *A.Levitin, Introduction to the Design and Analysis of Algorithms, Pearson Education, New Delhi, 2012.*
3. T.H.Cormen, C.E. Leiserson, and R.L. Rivest, Introduction to Algorithms, MIT Press, Cambridge, MA 1992.