Randomized Algorithms

A randomized Algorithm is an algorithm that employs a degree of randomness as part of its logic. The algorithm typically uses uniformly random values as an input to guide its behavior, in the hope of achieving good performance in the average case over all possible choices of random values.

An Algorithm that uses random numbers to decide what to do next anywhere in its logic, is called randomized algorithm, for example in randomized quick sort, we decide the next pivot at random.

There are two main types of theses algorithms.

1) Las Vegas

Algorithm that uses the random input so that they always terminate with the correct answer, but where the expected running time is finite.

2) Monte Carlo

Algorithm which have a chance of producing an incorrect result, but the run time is fixed.

For example: **Randomized Search**

We have an array A, which is the next one (1,2,3,4,5,6,7,8,9) and we want to search n = 6

Randomized Las Vegas Search:

```
def LasVegas (Array, n):
        while True:
                if Array[random] == n:
                        return Position
```

Randomized Monte Carlo Search:

```
def MonteCarlo (Array, n, loops):
        for n in range (1, loops):
                if Array[random] == n:
                        return Position
                else:
                        return Not found.
```

In these examples, Las Vegas Algorithm will run until found the correct answer. But its running time is totally random.

The Monte Carlo Algorithm will be complete in an amount of time that can be bounded to a function depending on the input size and the parameter of searching, but has a probability to fail.