**Problem Statement:** Write a client that takes integers M, N, and T as command-line arguments, then uses the code given in the text to perform T trials of the following experiment: Generate N random int values between 0 and M – 1 and count the number of duplicates. Run your program for T = 10 and N = 10^3, 10^4, 10^5, and 10^6, with M = N/2, and N, and 2N. Probability theory says that the number of duplicates should be about (1 – e^(–alpha) ) where alpha = N/M—print a table to help you confirm that your experiments validate that formula.

**Solution:**

By Using the formula (1 - e^(-alpha)) and running constant arguments with M, N and 2N considering the build type to be in the range of (10^3 - 10^6). We attain the following result.

**Results**

Values Generated | Max Value | Distinct Values | Expected Distinct Values

1000 500 435.10 432.33

1000 1000 631.90 632.12

1000 2000 792.10 786.94

10000 5000 4320.90 4323.32

10000 10000 6318.00 6321.21

10000 20000 7875.50 7869.39

100000 50000 43246.40 43233.24

100000 100000 63284.50 63212.06

100000 200000 78689.10 78693.87

1000000 500000 432363.20 432332.36

1000000 1000000 632030.00 632120.56

1000000 2000000 786961.40 786938.68

**Findings:** This is a naive approach where we are supposed to insert the values based on the index pair of the element. Then compute the mean of that values and check with the actual result.

**Complexity:** The worst case will be O(T\*(M + N)), The space required to compute is 3 arrays with sizes of M,N,T.

**Problem Statement:** Redo Exercise 2.5.31 using the Dedup filter given on page 490.

Compare the running times of the two approaches. Then use Dedup to run the experiments for N = 10^7, 10^8, and 10^9, repeat the experiments for random long values and discuss the results.

**Solution:** Here, I had used HashSet where no multiple values are allowed and no null values are encouraged. It maintains no order. We attain the following result. I had used dedup method and randomly generated keys. Due to hardware issues I had computed till 10^7.

**Results**

Method | Values type | Values Generated | Max Value | Time spent

DeDup Long 10000000 5000000 25.69

DeDup Long 10000000 10000000 29.37

DeDup Long 10000000 20000000 33.38

**Findings:** This method is far more time consuming as we compared with our basic approach it doubles the time as we had given long values to compute.

**Complexity:** The Time taken in worst case is O(N) where N represents the values generated.For adding and checking an integer O(1).

**Problem Statement:** Redo Exercise 3.5.30 using StringSET (see Exercise 5.2.6) instead of HashSET. Compare the running times of the two approaches. Then use Dedup to run the experiments for N = 10^7, 10^8, and 10^9, repeat the experiments for random long values and discuss the results.

**Solution:** Due to hardware compatbilties I had used at max 10^7 which makes the to attain the results.

**Result:**

Method | Data structure | Values type | Values Generated | Max Value | Time spent

DeDup StringSet(R-way trie) Long 10000000 5000000 74.32

DeDup StringSet(R-way trie) Long 10000000 10000000 83.83

DeDup StringSet(R-way trie) Long 10000000 20000000 92.99

Dedup StringSet(TST) Long 10000000 50000000 22.62

Dedup StringSet(TST) Long 10000000 10000000 27.82

**Findings:** This approach remains more time consuming R-way trie used more space as compared to our Trieset. Trieset performs well in this case as it holds out with the old instance and computes the values.

**Complexity:** The Worst case in R-way tries can be considered in case of search hit O(constant) and search miss is O(logN)R an to add an element it consumes O(constant). The space required to compute the process is (R+1)N. The worst case in TST can be considered in case of search hit O(ln) and search miss is O(ln N) and to add an element it consumes (ln + N) . The space required to compute the process is 4N.

**Conclusion:** Based on the above findings I would like to conclude that StringSet(TST) a bit better than usual Hashset as it can compute in a logarithmic time.