Randomized Algorithms and Linear time Sorting

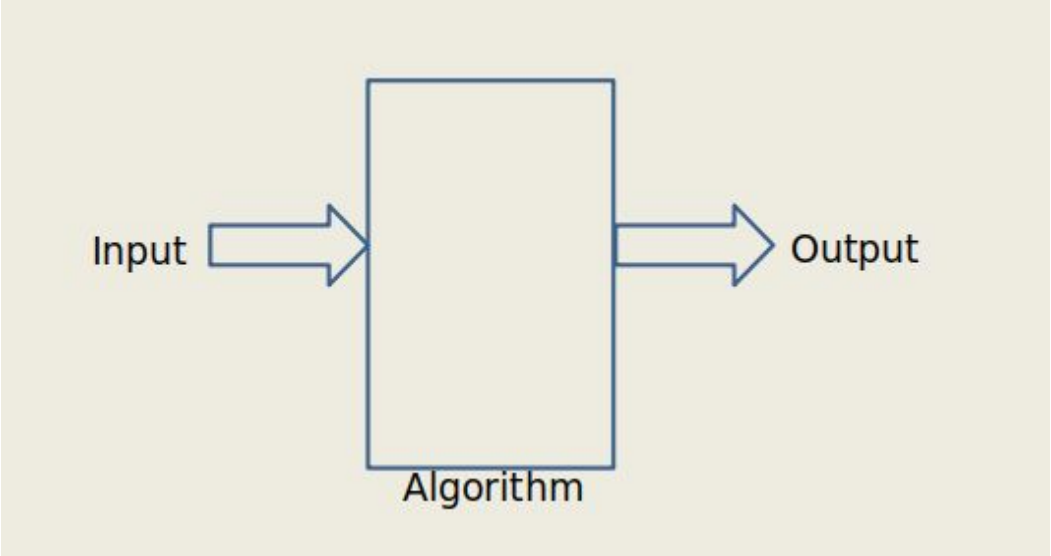
By : Ansari M.Saeem
UID:2019430001

Overview

- Deterministic Algorithm
- Randomized Algorithm
  - Monte Carlo
  - Las Vegas
- Example of Randomized Algorithm (Hiring Assistant Problem)
- Linear Sorting Algorithm
  - Counting Sort
  - Radix Sort
  - Bucket Sort

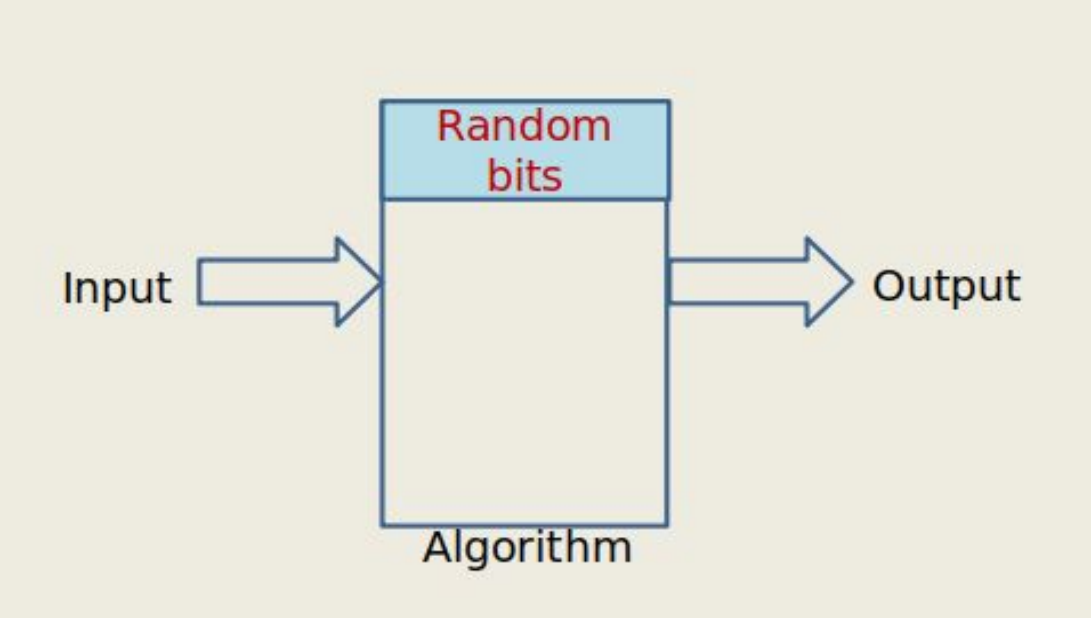
Deterministic Algorithm

- Before going to understand Randomized algorithm let's first see what is Deterministic algorithm.
- Output is always same for Same input.



Randomized Algorithms

- Output is different for same input.



Randomized Algorithms(continued)

- Two Category :
  - Monte Carlo
  - Las vegas
- Example :
  - Hiring Assistant Problem
  - Picking a random no from array as pivot in Quicksort algorithm.
- Application:
  - Cryptography
  - Data Structures
  - Graph Algorithm

Randomized Algorithm : Monte Carlo Algorithm

- May produce result based on some probability.
- Result may or may not be correct with some margin error.
- Runs for a fixed no of steps.
- Application :Physics, game theory, and finance
- Used in Karger’s Algorithm

Randomized Algorithm : Las Vegas Algorithm

- Always produce correct or optimum result.
- Runs until find a correct result.
- Accuracy of Las Vegas is higher than Monte Carlo.
- Used in Randomized Quick-Sort Algorithm.

Example of Randomized Algorithm : Hiring Assistant Problem

- Taking interview of one candidate per day
- Cost of conducting interview is c_i
- Cost of hiring one candidate is c_h
- $c_h > c_i$
- Candidate i is hired iff candidate i is better than 1,2,....,i-1
- Worst case : $O(c_i n + c_h) = O(c_i n)$
- Best case : $O(c_i n + c_h n) = O(c_t n)$
- Average case : ????

```
Hire(n)

    best = 0
    For i=1 in n
    Interview candidate i
        If candidate i is better than best
            Best = i
            hire candidate i
```

Example of Randomized Algorithm : Randomized Hiring Assistant Problem

- Get the list of candidate first
- Randomly select the candidate
- Candidate i is hired iff candidate i is better than 1,2,....,i-1
- P(candidate i is the hired) = 1/i
- Complexity : $O(c_h \ln n)$

```
Hire(n)

    Randomly permute the list of candidate
    best = 0
    For i=1 in n
        Interview candidate i
        If candidate i is better than best
            Best = i
            hire candidate i
```

Linear Sorting Algorithms

- Does not required element comparison
- Have complexity O(n)
- Run faster than O(n ln n)
- Counting sort, radix sort and bucket sort
- Not very desirable from practical point of view :
  - Efficiency depend on the random ordered.
  - Require extra space proportional to the size of the array being sorted
  - Even though they are linear, they would not be as faster than other sort(say Quick-Sort).

Example of linear Sorting Algorithm : Counting sort(steps)

- Initialize the auxiliary array Aux[] as 0.
  - Note: The size of this array should be ≥max(A[]).
- store the count of occurrence of each element in the appropriate index of the Aux array
- Traverse array Aux and copy i into new sorted Array for Aux[i](no of time element has repeated) number of times

Counting Sort (continued)

Input Data

0	4	2	2	0	0	1	1	0	1	0	2	4	2

Count Array

0	1	2	3	4
5	3	4	0	2

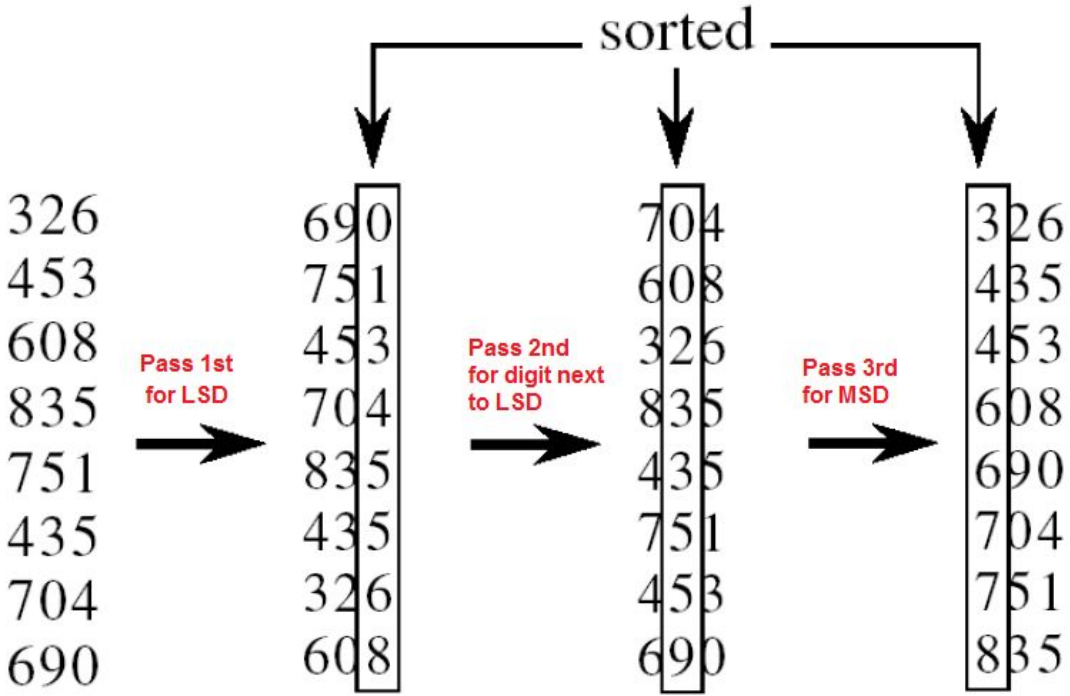
Sorted Data

0	0	0	0	0	1	1	1	2	2	2	2	4	4

Example of linear Sorting Algorithm : Radix Sort(steps)

1. Define 10 queues each representing a bucket for each digit from 0 to 9.
2. Consider the least significant digit of each number in the list.
3. Insert each number into their respective queue based on the least significant digit.
4. Group all the numbers from queue 0 to queue 9 in the order they have inserted into their respective queues.
5. Repeat from step 3 based on the next least significant digit.
6. Repeat from step 2 until all the numbers are grouped based on the most significant digit.

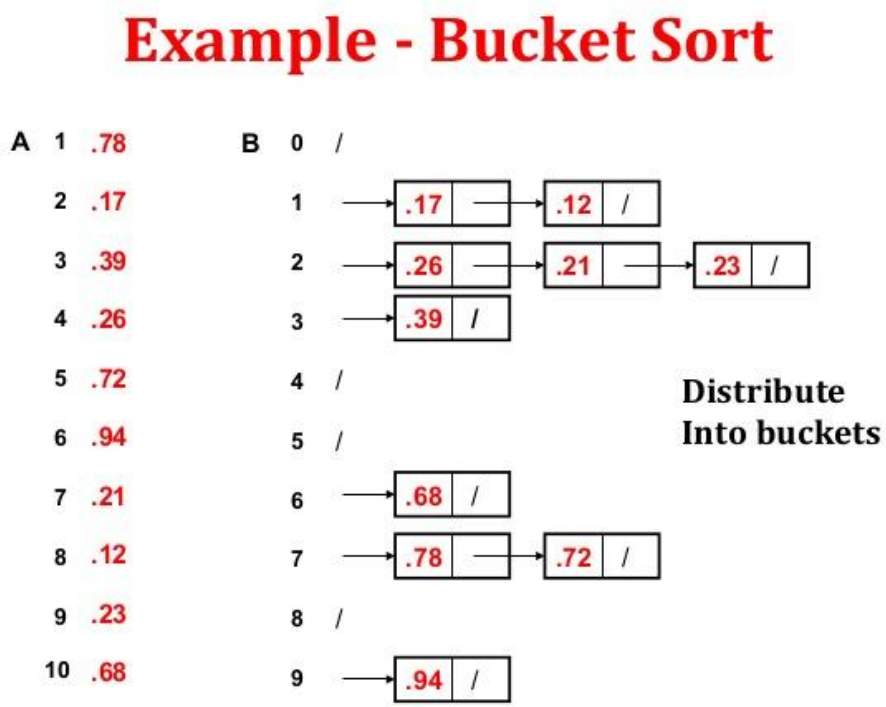
Radix Sort(continued)



Example of linear Sorting Algorithm : Bucket Sort(steps)

- Create n empty buckets (Or lists).
- Do following for every array element arr[i].
  - Insert arr[i] into bucket[n*array[i]]
- Sort individual buckets using insertion sort.
- Concatenate all sorted buckets.

Bucket Sort


Example - Bucket Sort
A
1 .78
2 .17
3 .39
4 .26
5 .72
6 .94
7 .21
8 .12
9 .23
10 .68
B
0 /
1 .17 .12 /
2 .26 .21 .23 /
3 .39 /
4 /
5 /
6 .68 /
7 .78 .72 /
8 /
9 .94 /
Distribute
Into buckets

Reference Paper : LINEAR TIME SORTING AND CONSTANT TIME SEARCHING ALGORITHMS

- Time Complexity : O(n)
- At the time of insertion, insert element (say Data) such that array will be sorted.

Algorithm:

```
while not sorted do
    if A[i] == 0 then
        A[i] == Data
        Exit loop
    Else if A[i] < Data then
        i = i+1
    Else
        temp = A[i]
        A[i] = Data
        Data = temp
        i = i+1
```

Reference

- INTRODUCTION TO ALGORITHMS THIRD EDITION by THOMAS CORMEN

  CHARLES E. LEISERSON RONALD L. RIVEST CLIFFORD STEIN

- Randomized Algorithms by Prabhakar Raghavan ,IBM Almaden Research Center ,San Jose, CA.
- Paper:PS911O LINEAR TIME SORTING AND CONSTANT TIME SEARCHING ALGORTHMS
- www.geeksforgeeks.org
- www.hackerearth.com
- www.slideshare.net
- www.freecodecamp.org

Thank You