Substituting (4) and (5) into (3) and using $p_i = \frac{1}{n-2}$ yields

$$E\{N_i^2\} = 2 - \frac{1}{n-2} \qquad (6)$$

Substituting (6) into (2) we have

$$\begin{aligned} E\{T(n)\} &= c\sum_{i=1}^{n-2}\left(2 - \frac{1}{n-2}\right) \\ &= 2cn - 5c \\ &= O(n) - O(1) \\ &= O(n) \end{aligned}$$

Therefore, for points uniformly distributed in the unit interval, algorithm BUCKET-SORT runs in linear expected time.

$$T(n) = k_1 N_1^2 + k_1 N_2^2 + \dots + k_{n-2} N_{n-2}^2$$

$$= c \sum_{i=1}^{n-2} N_i^2 \qquad (1)$$

where c is a positive constant.

To find the expected time we need to take the expected value, denoted by $E\{\bullet\}$, of (1).

$$E\{T(n)\} = c \sum_{i=1}^{n-2} E\{N_i^2\} \qquad (2)$$

Thus we need to know the expected value of the square of a random variable. Now, for any random variable X we have

$$E\{X^2\} = \mu^2 + Var(X) \qquad (3)$$

This is easy to see from the definition of the variance since

$$Var(X) = E\{(X - \mu)^2\}$$

$$= E\{X^2 - 2\mu X + \mu^2\}$$

$$= E\{X^2\} - 2\mu E\{X\} + \mu^2$$

$$= E\{X^2\} - \mu^2$$

Furthermore, for a binomial random variable N_i with parameters (n-2) and p_i we have that:

$$\mu = (n-2)p_i \qquad (4)$$

and

$$Var(X) = (n-2)p_i(1-p_i) \qquad (5)$$

Bucket Sorting in O(*n*) Expected Time

Godfried Toussaint

School of Computer Science
McGill University
Montreal, Quebec, Canada

1. Introduction

Given n numbers $X_1, X_2, ..., X_n$ drawn at random independently from the uniform distribution in [0,1], it is desired to sort them in O(n) expected time.

Our model of computation allows the floor function to be performed in constant time. The following algorithm does the job.

2. Algorithm BUCKET-SORT

```
Begin
    Step 1: Find $X_{min}$ and $X_{max}$, the points with minimum and maximum value.
    Step 2: Divide the interval [$X_{min}$, $X_{max}$] into $n$-2 “buckets” or intervals of equal length.
    Step 3: “Throw” the remaining $n$-2 points into their respective buckets using the floor function.
    Step 4: For each bucket that contains more than one point sort them with any method that runs
            in at most quadratic worst-case time.
    Step 5: Scan through the buckets and concatenate the sorted lists in each bucket.
End
```

3. Analysis

Once X_{min} and X_{max} are found the algorithm processes the remaining n-2 points which are themselves *uniformly* distributed in [X_{min}, X_{max}]. Since we have n-2 buckets it follows that the probability that a remaining point falls in the *i-th* bucket is $p_i = 1/(n\text{-}2)$. In other words, the number of points that falls in bucket i is a *binomial* random variable, denoted by N_i, with parameters (n-2) and p_i, i = 1, 2,..., n-2. If we sort each N_i using a quadratic time algorithm the total time taken by BUCKET-SORT is given by