The **Maximum Gap Problem**: An Algorithmic Application of the **Pigeonhole Principle**

Godfried Toussaint
The Maximum Gap Problem: An Algorithmic Application of the Pigeonhole Principle

A linear-time algorithm for computing the maximum gap allowing the constant time computation of floor functions in the model of computation.

Given a set $S$ of $n > 2$ real numbers $x_1, x_2, ..., x_n$.

1. Find the maximum, $x_{-max}$ and the minimum, $x_{-min}$ in $S$.

2. Divide the interval $[x_{-min}, x_{-max}]$ into $(n-1)$ "buckets" of equal size $\delta = (x_{-max} - x_{-min})/(n-1)$.

3. For each of the remaining $n-2$ numbers determine in which bucket it falls using the floor function. The number $x_i$ belongs to the $k$th bucket $B_k$ if, and only if, $\lfloor (x_i - x_{-min})/\delta \rfloor = k-1$.

4. For each bucket $B_k$ compute $x_{k-min}$ and $x_{k-max}$ among the numbers that fall in $B_k$. If the bucket is empty return nothing. If the bucket contains only one number return that number as both $x_{k-min}$ and $x_{k-max}$.
The Maximum Gap Problem: An Algorithmic Application of the Pigeonhole Principle

5. Construct a list L of all the ordered minima and maxima:
   \[ L: (x_1\text{-min}, x_1\text{-max}), (x_2\text{-min}, x_2\text{-max}), \ldots, (x_{n-1}\text{-min}, x_{n-1}\text{-max}), \]

   • Note: Since there are \(n-1\) buckets and only \(n-2\) numbers, by the Pigeonhole Principle, at least one bucket must be empty. Therefore the maximum distance between a pair of consecutive points must be at least the length of the bucket. Therefore the solution is not found among a pair of points that are contained in the same bucket.

6. In L find the maximum distance between a pair of consecutive minimum and maximum \((x_i\text{-max}, x_j\text{-min})\), where \(j > i\).

7. Exit with this number as the maximum gap.