Bucket Sort

Code

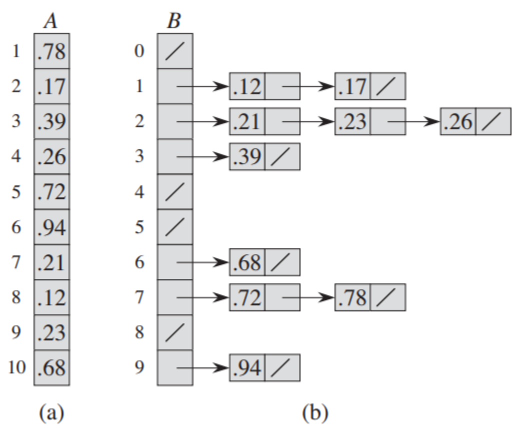
```
BUCKET-SORT(A)
let B[0..n − 1] be a new array
n = A.length
for i = 0 to n − 1
    make B[i] an empty list
for i = 1 to n
    insert A[i] into list B[⌊nA[i]⌋]
for i = 0 to n − 1
    sort list B[i] with insertion sort
concatenate the lists B[0], B[1], ..., B[n − 1] together in order
```

Design

- bucket sort assumes the input has elements evenly distributed over the interval 0 and 1
- divide the interval into n equal sized sub-interval buckets in an array B
  - where each element in B is the head of a linked list (i.e. a bucket)
- distribute the input elements into the buckets
  - for a bucket i, it covers the domain of $[i \times \frac{1}{n}, (i+1)\frac{1}{n}]$
  - if an element has a value a, its bucket index is

$$
\begin{gathered}
i \times \frac{1}{n} \leq a \leq (i+1)\frac{1}{n} \\
i \leq a * n \leq i + 1 \\
i = \lfloor a \times n \rfloor
\end{gathered}
$$

- sort each bucket with insertion sort
- go through each bucket to list the elements as a sorted array



Runtime Analysis

Worst Case

In the worst case, all the elements are placed in the same bucket and the runtime is $O(n^2)$

Average Case

$$T(n) = \Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)$$

Where n_i is the number of elements that fall into bucket i. We will use expectation for the average case.

$$\begin{aligned} E[T(n)] &= E[\Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)] \\ &= \Theta(n) + \sum_{i=0}^{n-1} O(E(n_i^2)) \end{aligned}$$

Use X_{ij} as a RV that `A[j]` falls into bucket i.

$$X_{ij} = \begin{cases} 0 & \text{with probability } 1 - \frac{1}{n} \\ 1 & \text{with probability } \frac{1}{n} \end{cases}$$

49:33 L11