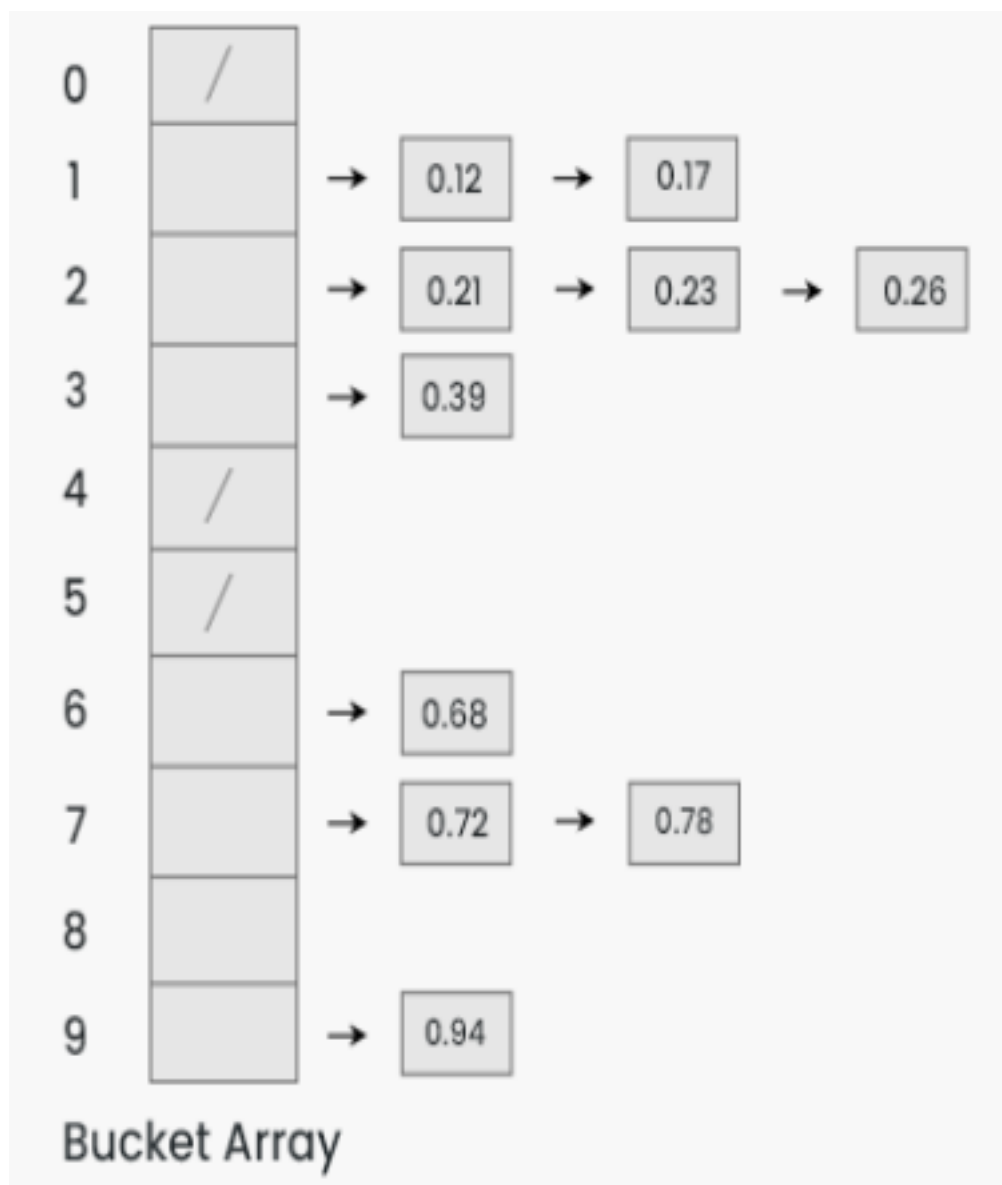

Ipotesi: > Vettore A di numeri $\in \mathbb{R}$ in $[0, 1[$ distribuiti uniformemente.

Gli elementi di A sono ottenuti estraendo dall'intervallo $[0, 1]$ con distribuzione uniforme.

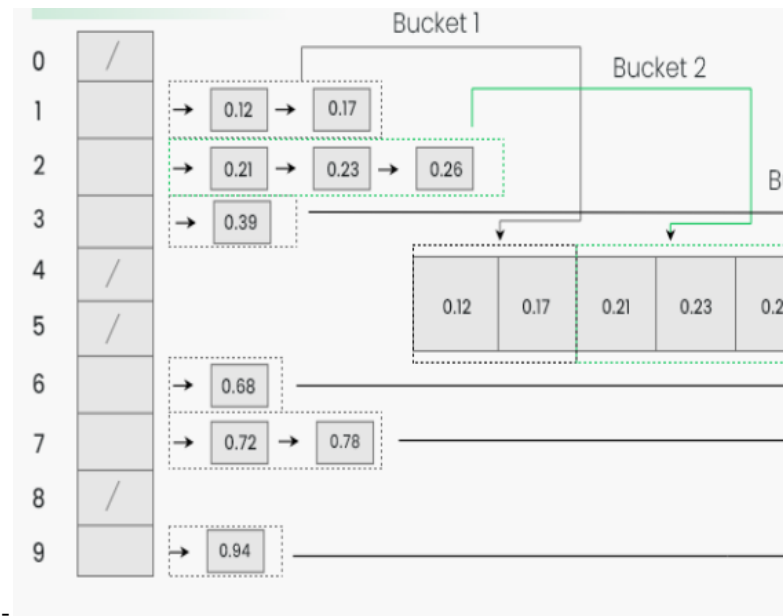
Procedimento

Step 1 1. Si crea un vettore ausiliario C (bucket array) della stessa dimensione di A. 2. Si moltiplica l'elemento per la dimensione di C e si converte il risultato in intero. 3. Si inserisce l'elemento nel bucket corrispondente all'indice calcolato 4. Repeat for each element in A.

Step 2 1. Ordinare gli elementi dentro ogni bucket (qualsiasi algoritmo di ordinamento) 2. In media vengono ordinate $\Theta(n) \cdot \Theta(1)$ liste -> quindi la complessità media è $\Theta(n)$, se A ha distribuzione uniforme. Il caso peggiore è dato dall'algoritmo di ordinamento scelto, quindi $\Theta(n^2)$ per insertion sort.



Step 3 - Dove in-



serieA[i]? - insertList(C[A[i]-n], A[i]) -

The array is now sorted.

[!attention]- Code

```

1 bucketSort(A, n){
2   let B[0:n-1] // new array, same dimension as A
3   for(i=0 to n-1){
4     make B[i] an empty list
5   }
6   for(i=1 to n){
7     insert A[i] into list B[n*A[i]] // parte intera inferiore
8   }
9   for (i=0 to n-1){
10    sort list B[i] // insertion sort
11  }
12  concatenate the lists B[i] together in order
13  return concatenated lists
14 }

```