

# BUCKET SORT

## GRUPO AMRV

E.Aguila M. Marabolí P. Rain B. Vásquez

Universidad Tecnológica Metropolitana

26 de Julio del 2013



# INTRODUCCIÓN

El estudio de algoritmos de ordenamiento tiene una gran importancia dentro de la Ciencia de la Computación, pues una buena cantidad de los procesos realizados por medios computacionales requieren que sus datos estén ordenados. Además, el hecho de almacenar los datos de manera ordenada permite implementar algoritmos de búsqueda muy rápidos (por ejemplo: búsqueda binaria). Esta y muchas otras razones de practico impulsaron el estudio y la búsqueda de algoritmos de ordenamiento recientes.



# CARACTERISTICAS

- ▶ Corre en tiempo lineal cuando la entrada se toma de una distribución uniforme.
- ▶ Rápido porque asume algo sobre la entrada (como counting sort, asume números en un rango pequeño).
  - ▶ Bucket Sort asume números generados aleatoriamente y distribuidos uniformemente en un rango de  $[0,1)$ .
- ▶ Divide el intervalo de  $[0,1)$  en  $n$  sub-intervalos de igual tamaño.
  - ▶ Se ordenan los números en cada partición.
  - ▶ Se recorren las particiones en orden listando los elementos.



# ALGORITMO

```
función bucket-sort(elementos, n)
casilleros  colección de n listas
para i = 1 hasta longitud(elementos) hacer
  c  buscar el casillero adecuado
  insertar elementos[i] en casillero[c]
fin para
para i = 1 hasta n hacer
  ordenar(casilleros[i])
fin para
devolver la concatenación de casilleros[1],..., casilleros[n]
```



# COMPLEJIDAD

Para un arreglo de largo  $n$  y  $k$  Buckets.

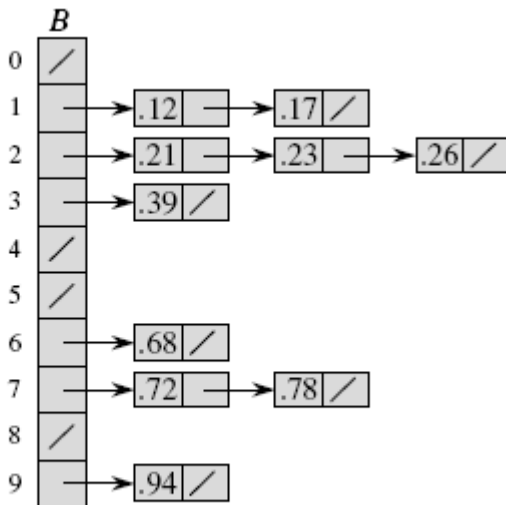
- ▶ Peor Caso:  $O(n^2)$
- ▶ Caso Promedio:  $O(n + k)$
- ▶ Complejidad Espacial:  $O(nk)$



# EJEMPLO

<i>A</i>	
1	.78
2	.17
3	.39
4	.26
5	.72
6	.94
7	.21
8	.12
9	.23
10	.68

(a)



(b)

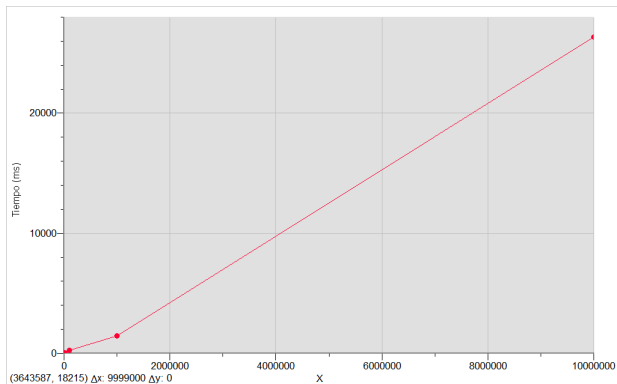


# PROBLEMAS

- ▶ Cuanto más reducido sea el rango de valores que pueden tomar los casilleros y mayor sean los datos del conjunto, más desequilibrio existirá. Si hay desequilibrio, ordenara particiones de tamaños diferentes.
- ▶ El rendimiento depende del tamaño y sobre todo la distribución que posean los datos que se estén ordenando.
- ▶ Esto puede usarse a favor cuando se están ordenando elementos de estructuras lineales, como las listas.



# RESULTADOS





# CONCLUSIONES

- ▶ Una ventaja de Bucket sort con respecto a todos los demás es que, si se puede implementar, entonces es el método más rápido que existe para ordenar listas muy grandes.
- ▶ Si los datos están muy desequilibrados puede desencadenar en un mal rendimiento del algoritmo.

