Teorema. *Sejam X e Y v.a.'s independentes. Então* $\mathrm{E}[XY] = \mathrm{E}[X] \cdot \mathrm{E}[Y]$.

Bucketsort

Prof. André Vignatti

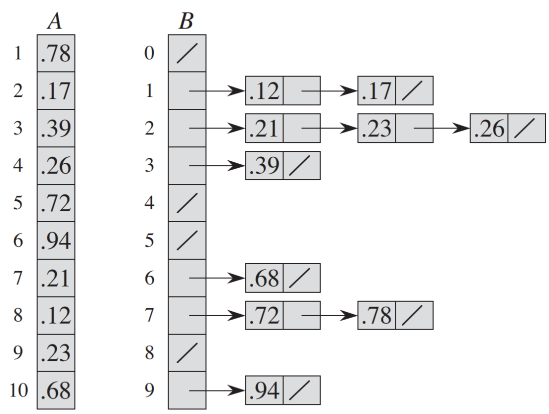
Suposição: entrada com distribuição aleatória uniforme dos elementos em $[0, 1)$.

Idéia:

- Divida $[0, 1)$ em n baldes de tamanhos iguais.
- Distribua os n valores de entrada nos baldes.
- Ordene cada balde.
- Percorra os baldes em ordem, listando os elementos em cada um.

BucketSort(A,n)

seja $B[0..n-1]$ um novo vetor;
for $i = 1$ **to** $n - 1$ **do**
    faça $B[i]$ uma lista vazia
for $i = 1$ **to** n **do**
    insira $A[i]$ na lista $B[\lfloor n \cdot A[i] \rfloor]$
for $i = 1$ **to** $n - 1$ **do**
    ordene a lista $B[i]$ com InsertionSort
return *a concatenação das listas* $B[0], B[1], \ldots, B[n-1]$ *em ordem*



Corretude

Considere $A[i]$ e $A[j]$. Assuma sem perda de generalidade que $A[i] \leq A[j]$. Então $\lfloor n \cdot A[i] \rfloor \leq \lfloor n \cdot A[j] \rfloor$. Então $A[i]$ é colocado no mesmo balde que $A[j]$ ou em um balde com um índice menor.

- Se for no mesmo balde, a ordenação por inserção corrige.
- Se for no balde anterior, a concatenação corrige.

Tempo Esperado

- Todas as linhas do algoritmo, exceto a ordenação por inserção, leva $\Theta(n)$.
- Se cada balde recebe um número constante de elementos: $O(1)$ para ordenar cada balde $\Rightarrow O(n)$ para ordenar tudo.
- Esperamos que cada balde tenha poucos elementos, já que a média é 1 elemento por balde.
- Mas precisamos fazer uma análise cuidadosa.

Defina uma variável aleatória:

- n_i: número de elementos colocados no balde i.

Como o InsertionSort executa em tempo quadrático, então o tempo do BucketSort é:

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

E o valor esperado é

$$\begin{aligned}
\mathrm{E}[T(n)] &= \mathrm{E}\left[\Theta(n) + \sum_{i=0}^{n-1} O(n_i^2)\right] \\
&= \Theta(n) + \sum_{i=0}^{n-1} \mathrm{E}[O(n_i^2)] \qquad \text{(linearidade da esperanca)} \\
&= \Theta(n) + \sum_{i=0}^{n-1} O(\mathrm{E}[n_i^2]) \qquad (\mathrm{E}[aX] = a\mathrm{E}[X])
\end{aligned}$$

Teorema. $\mathrm{E}[n_i^2] = 2 - 1/n$ *para* $i = 0, \ldots, n-1$.

Demonstração. Seja X_{ij} v.a. tal que $X_{ij} = 1$ se $A[j]$ cai no balde i, e $X_{ij} = 0$ c.c. Desta forma,

$$n_i = \sum_{j=1}^{n} X_{ij}.$$

Assim,

$$\begin{aligned}
\mathrm{E}[n_i^2] &= \mathrm{E}\left[\left(\sum_{j=1}^{n} X_{ij}\right)^2\right] \\
&= \mathrm{E}\left[\sum_{j=1}^{n}\sum_{k=1}^{n} X_{ij}X_{ik}\right] \\
&= \mathrm{E}\left[\sum_{j=1}^{n} X_{ij}^2 + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n\\ k\ne j}} X_{ij}X_{ik}\right] \\
&= \sum_{j=1}^{n} \mathrm{E}[X_{ij}^2] + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n\\ k\ne j}} \mathrm{E}[X_{ij}X_{ik}].
\end{aligned}$$

Note que $\Pr(X_{ij}=1)=\frac{1}{n}$, portanto

$$\mathrm{E}[X_{ij}^2] = 1^2\cdot\Pr(X_{ij}^2=1) + 0^2\cdot\Pr(X_{ij}^2=0) = 1^2\cdot\Pr(X_{ij}=1) = \frac{1}{n}.$$

Quando $k \ne j$, as v.a. X_{ij} e X_{ik} são independentes, então

$$\mathrm{E}[X_{ij}X_{ik}] = \mathrm{E}[X_{ij}]\mathrm{E}[X_{ik}] = \frac{1}{n}\cdot\frac{1}{n} = \frac{1}{n^2}.$$

$\square$

Substituindo os valores encontrados,

$$\begin{aligned}
\mathrm{E}[n_i^2] &= \sum_{j=1}^{n}\frac{1}{n} + \sum_{1\le j\le n}\ \sum_{\substack{1\le k\le n\\ k\ne j}} \frac{1}{n^2} \\
&= n\cdot\frac{1}{n} + n(n-1)\cdot\frac{1}{n^2} \\
&= 1 + \frac{n-1}{n} \\
&= 2 - \frac{1}{n}.
\end{aligned}$$

Portanto,

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

Lembrando: não é uma ordenação por comparação.

- Esta é uma análise probabilística - usamos probabilidade para analisar um algoritmo cujo tempo de execução depende da distribuição de entradas.
- Diferente de um algoritmo aleatorizado, onde usamos a aleatorização para impor uma distribuição.