



# Ejercicios del primer set de slides

1. Número de runs
2. Carga máxima variable en separate chaining
3. Quadratic probing
4. Uniform Hashing
5. Linear probing

# Problema: número de runs

```
def runs(arr): # arr = permutacion
    res = []
    i, n = 0, len(arr)
    while i < n:
        j = i + 1
        if j < n and arr[i] <= arr[j]:
            # creciente
            while j < n and arr[j - 1] <= arr[j]:
                j += 1
        elif j < n and arr[i] > arr[j]:
            # decreciente
            while j < n and arr[j - 1] > arr[j] :
                j += 1
        else:
            # elemento aislado
            j = i + 1
        res.append(j - i)
        i = j
    return res
```

# Problema: número de runs

## Problema

La cantidad esperada de runs es  $\mathbb{E}[r] \sim cn$  para una cierta  $c > 0$ .

Veamos la permutación como una secuencia  $X_1, X_2, \dots$  de números iid de  $[0, 1]$ .

- (a) Probar  $runs(X_1, \dots, X_{i+j}) \leq runs(X_1, \dots, X_i) + runs(X_{i+1}, \dots, X_{i+j})$ .
- (b) Probar que  $e_k := \mathbb{E}[runs(X_1, \dots, X_k)]$  satisface  $e_{i+j} \leq e_i + e_j$  para todo  $i, j \geq 0$ . Concluir que  $e_k/k \rightarrow c$  para cierta  $c \geq 0$ .
- (c) Mostrar que la constante es positiva  $c > 0$ .

# Carga máxima variable en separate chaining

Consideremos una tabla de hash con  $K$  celdas y *separate chaining* (con listas encadenadas):

## Problema

¿Y si la carga máxima  $\gamma$  fuera una función  $\gamma = g(K)$  ?

Observar que la longitud media de las listas es  $\leq g(K)$ .

---

Notar que la capacidad de la tabla de hash es  $n \leq Kg(K)$ .

# Carga máxima variable en separate chaining

Consideremos una tabla de hash con  $K$  celdas y *separate chaining* (con listas encadenadas):

## Problema

¿Y si la carga máxima  $\gamma$  fuera una función  $\gamma = g(K)$  ?

Observar que la longitud media de las listas es  $\leq g(K)$ .

---

Notar que la capacidad de la tabla de hash es  $n \leq Kg(K)$ .

Recordamos que  $P_m := \Pr(\exists j : |C_j(n)| \geq m)$ , satisface

$$P_m \leq \exp(\log K + m \log g(K) + m - m \log m).$$

1. Considerar  $g(K) = \sqrt{\log K}$ . Probar que  $P_m \rightarrow 0$  para  $m = 2 \frac{\log K}{\log \log K}$ .
2. Considerar  $g(K) = \log K$ . Probar  $P_m \rightarrow 0$  para  $m = (\log K)^2$ .  
¿Para  $m = (\log K)$  podemos tener  $P_m \rightarrow 0$ ?

# Quadratic probing

Consideramos la secuencia de quadratic probing

$i_1 = i_0 + 1, i_2 = i_1 + 2, \dots, i_k = i_{k-1} + k, \dots$  módulo  $K$ , el tamaño.

## Ejercicio

Probar que si  $K = 2^m$  entonces  $i_0, i_1, \dots, i_{K-1}$  son distintos módulo  $K$ .

# Uniform Hashing

## Teorema (Búsqueda en Uniform hashing)

*El costo medio de una búsqueda con uniform hashing es*

$$U_n = \frac{K+1}{K-n+1} \sim \frac{1}{1-\alpha}, \quad S_n \sim \frac{1}{\alpha} \log \left( \frac{1}{1-\alpha} \right).$$

## Ejercicio

1. Probar que, para una búsqueda no exitosa<sup>a</sup>  $U_n = \frac{K+1}{K-n+1}$ .
2. Probar la fórmula para la búsqueda exitosa  $S_n$ .

---

<sup>a</sup>**Pista:** Inducción.



# Linear Probing

Para Linear Probing el costo de una búsqueda no-exitosa satisface

$$U_n \leq \frac{1}{2} \left( 1 + \frac{1}{(1 - \alpha)^2} \right)$$

## Ejercicio

Deducir que el costo medio de una búsqueda exitosa satisface

$$S_n \leq \frac{1}{2} \left( 1 + \frac{1}{1 - \alpha} \right).$$