Ejercicios del primer set de slides

1. Número de runs
2. Quadratic probing
3. Uniform Hashing

Problema: número de runs

```python
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 $X_1, X_2, \ldots$ de números iid de $[0,1]$.

(a) Probar $runs(X_1, \ldots, X_{i+j}) \leq runs(X_1, \ldots, X_i) + runs(X_{i+1}, \ldots, X_{i+j})$.

(b) Probar que $e_k := \mathbb{E}[runs(X_1, \ldots, X_k)]$ satisface $e_{i+j} \leq e_i + e_j$ para todo $i, j \geq 0$. Concluir que $e_k/k \to c$ para cierta $c \geq 0$.

(c) Mostrar que la constante es positiva $c > 0$.

[0] *Pista (b).* Lema de Fekete...

[0] *Pista (c).* ¿Qué podemos decir si $X_i < X_{i+1}$ y $X_{i+1} > X_{i+2}$?

Quadratic probing

Consideramos la secuencia de quadratic probing
$i_1 = i_0 + 1$, $i_2 = i_1 + 2, \ldots, i_k = i_{k-1} + k, \ldots$ módulo K, el tamaño.

Ejercicio

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

Uniform Hashing

Teorema (Búsqueda en Uniform hashing)

El costo medio de una búsqueda con uniform hashing es

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

Ejercicio

1. Sea X_n la cantidad de probes necesarios para realizar la n-ésima inserción. Notar que

$$U_n = \mathbb{E}[X_n] = \sum_{i=0}^{\infty} \Pr(X_n > i) = \sum_{i=0}^{n} \frac{\binom{n}{i}}{\binom{K}{i}}.$$

2. Probar que [a] $U_n = \frac{K+1}{K-n+1}$.
3. Probar la fórmula para la búsqueda exitosa S_n.

[a] **Pista:** Inducción.