

## TA3

### formas de obtener mediana:

1. orden  $n^2$  si hago burbuja y despues agarro el del medio ( $n^2 * \log N$  orden final)
2. Agarro el primero, medio y ultimo y hago el promedio. Orden 1 pero no es muy eficaz en el uso del algoritmo. si no están duplicados te asegura que no caes el peor caso. Si justo son el mismo entonces hago el mismo valor. Siempre el peor caso es que se repita.

### pseudocódigo:

```
function encuentraPivote(i, j):  
    a = A[i]  
    b = A[(i + j) // 2]  
    c = A[j]  
    if (a < b < c) or (c < b < a):  
        return (i + j) // 2  
    elif (b < a < c) or (c < a < b):  
        return i  
    else:  
        return j
```

3. Hago el promedio de los elementos  $O(N)$

```
public static int encuentraPivote(int[] A, int i, int j) {  
    // Calcular la suma de los elementos en el rango [i, j]  
    int suma = 0;  
    for (int k = i; k <= j; k++) {  
        suma += A[k];  
    }  
  
    // Calcular el promedio  
    double promedio = (double) suma / (j - i + 1);  
  
    // Encontrar el índice del elemento más cercano al promedio
```

```

int indicePivote = i;
double diferenciaMinima = Math.abs(A[i] - promedio);

for (int k = i + 1; k <= j; k++) {
    double diferenciaActual = Math.abs(A[k] - promedio);
    if (diferenciaActual < diferenciaMinima) {
        diferenciaMinima = diferenciaActual;
        indicePivote = k;
    }
}

return indicePivote;
}

```

4. El pivote es aleatorio: orden 1. lo acoto al i,j que estoy usando cada vez la probabilidad es mas grande  $n-2/n * n-2/n$  (num mas cerca de uno)
  - la probabilidad de caer en el peor caso es  $1/n$
  - la probabilidad de caer en un caso distinto del peor es  $n-2/n$ . esto es en una llamada de encuentra el pivote
  - Pseudocódigo:
 

```

function encuentraPivote(i, j):
    return random.randint(i, j)
                    
```