## Care este complexitatea următorului algoritm?

**a)**

```
s = 0
i = n
while i >= 1:
      j = 1
      while j <= n:
              s = s + 1
              j = j + 1
      i = i/2
```

**Soluție:**

Este suficient să presupunem $n=2^k$ (altfel putem lua n' cea mai mică putere a lui 2 care depășește n pentru a estima T(n) <= T(n'))

Pentru i=n se execută 2 instrucțiuni + cele două instrucțiuni din `while j <= n` se execută pentru j=1,2,.., n deci de **n** ori => ordin n operații (2n+2 operatii)

Pentru i=n/2 se execută 2 instrucțiuni + cele două instrucțiuni din `while j <= n` se execută pentru j=1,2,.., n deci de **n** ori => tot ordin n operații (2n+2 operatii)

Ultima valoare pentru i este $n/2^k=1$ (deci i ia in total k+1 = log(n) +1 valori ) => tot ordin n operații (2n+2 operatii)

- ⇨ T(n) = O(kn) = O(nlog(n)) (!!k=log2(n))

**b)**

```
s = 0
i = n
while i >= 1:
      j = 1
      while j <= i:
              s = s + 1
              j = j + 1
      i = i/2
```

**Soluție:**

Ca mai sus:

i=n => j=1,2,.., n => ordin n operații

i=n/2 => j=1,2,.., n/2 => ordin n/2 operații

....

=>$T(n) = O(n+n/2+n/2^2+...+n/2^k) = O(n(1+1/2+1/2^2+...+1/2^k)) = O(n)$ deoarece suma $1+1/2+1/2^2+...+1/2^k$ este <= 2 este suma de progresie geometrica = $2(1-1/2^{k+1})$