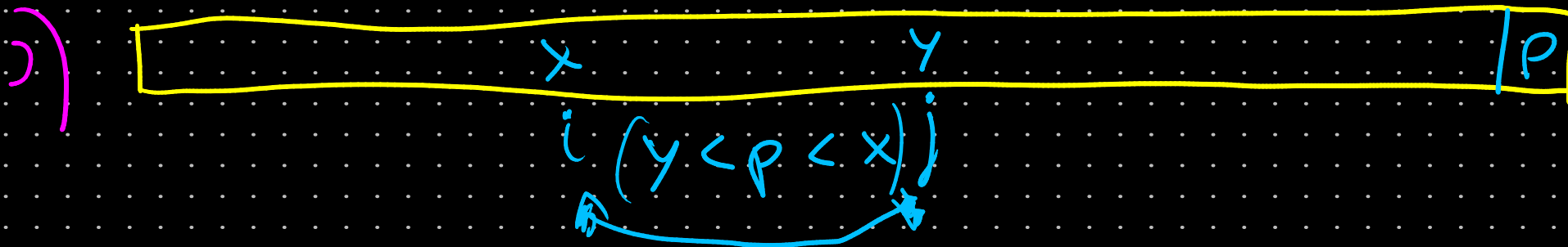
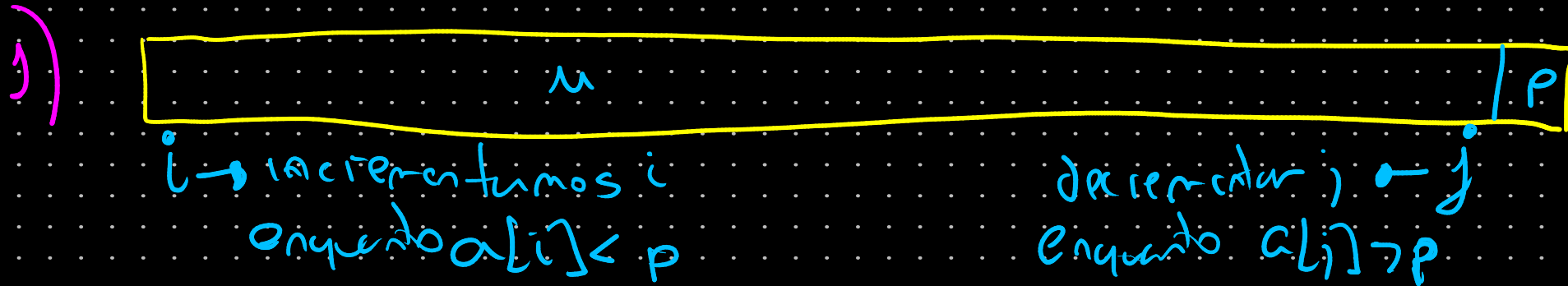
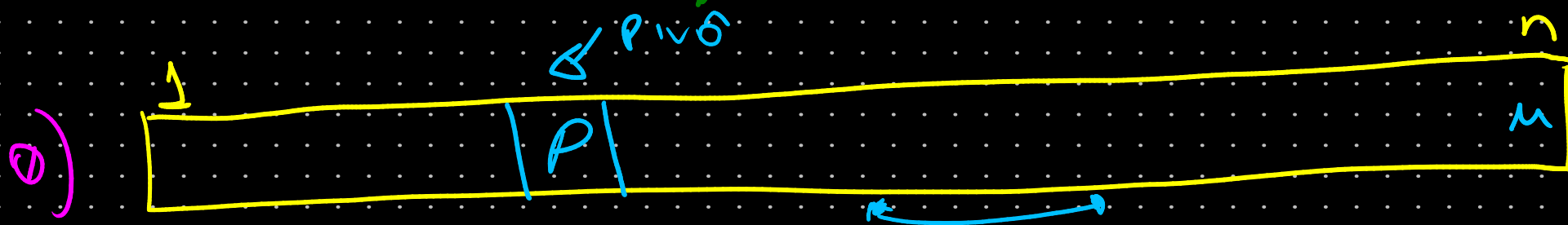
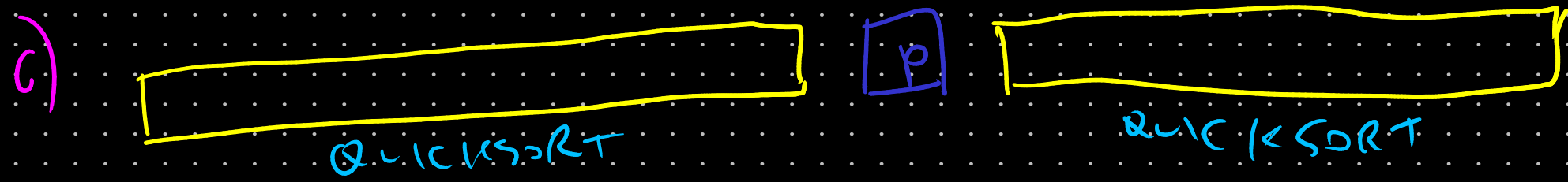
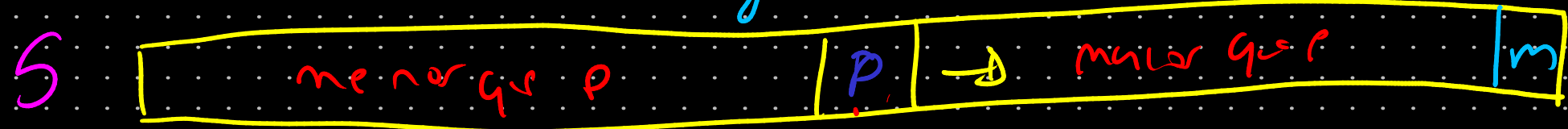
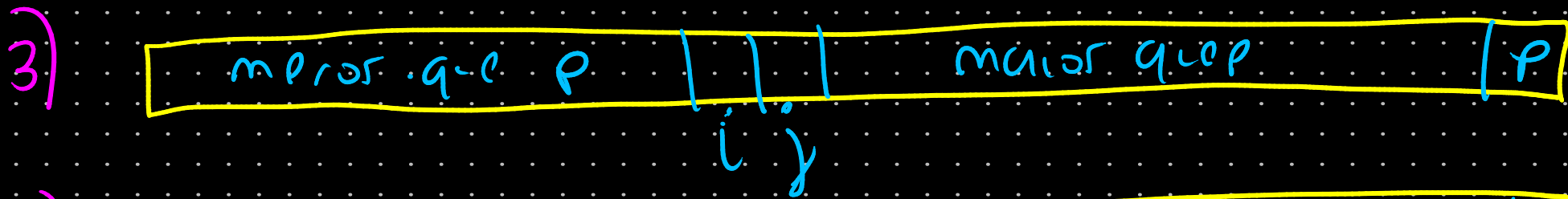


# Quicksort (Ordenação "Rápida")

- Recursivo (Dividir para conquistar)

Qual a ideia?





Algoritmo QuickSort(lista L, inicio, fim) {

$p \leftarrow \text{escolha\_pivot}(L)$ ;  $p \leftarrow L[p]$

  troca( $L[\text{fim}]$ ,  $L[p]$ )

$i \leftarrow \text{inicio}$

$j \leftarrow \text{fim} - 1$

  laço

    enquanto  $L[i] < p$  faça  $\text{Qn1}$

$i \leftarrow i + 1$

    enquanto  $L[j] \geq p$  faça  $\text{Qn1}$

$j \leftarrow j - 1$

    se  $i < j$  então

      troca( $L[i]$ ,  $L[j]$ )

    senão

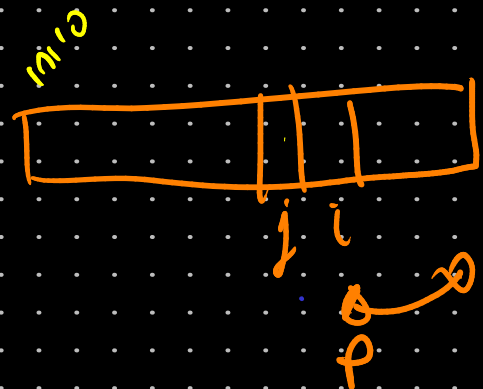
      troca( $L[i]$ ,  $L[\text{fim}]$ )

  QuickSort( $L$ ,  $\text{inicio}$ ,  $i$ )

  QuickSort( $L$ ,  $j + 1$ ,  $\text{fim}$ )

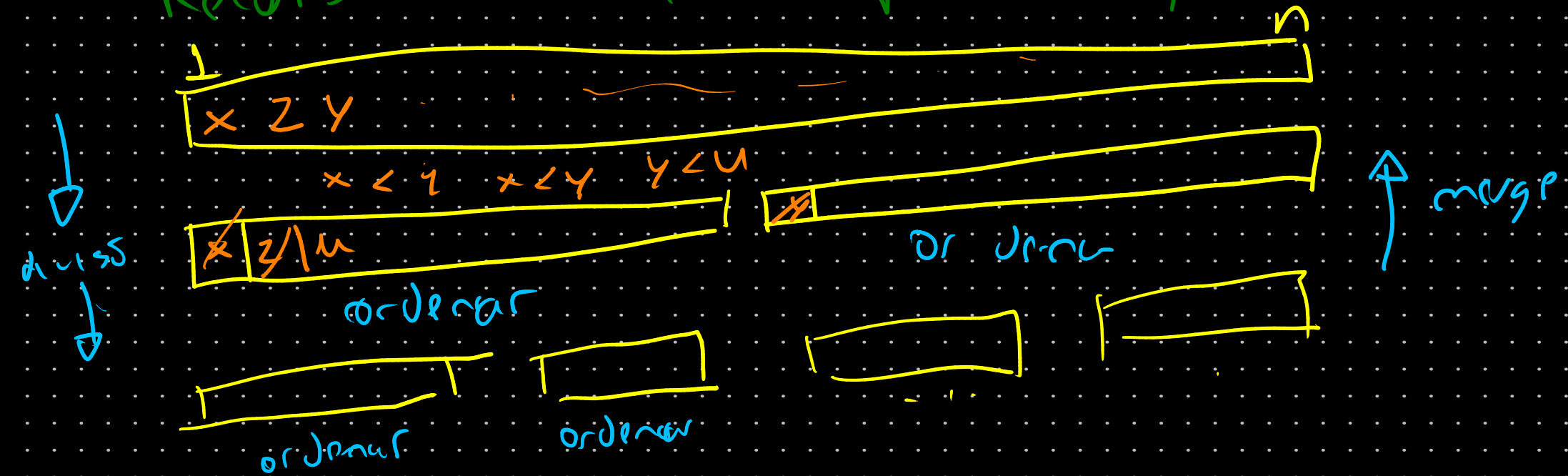
  fim do laço

  fim laço



# Mergesort (Ordenação por intercalação)

Recursão e Dividir para conquistar



Merge sort (Lista L, inicio, fin) 1 10

se inicio  $\neq$  fin entonces

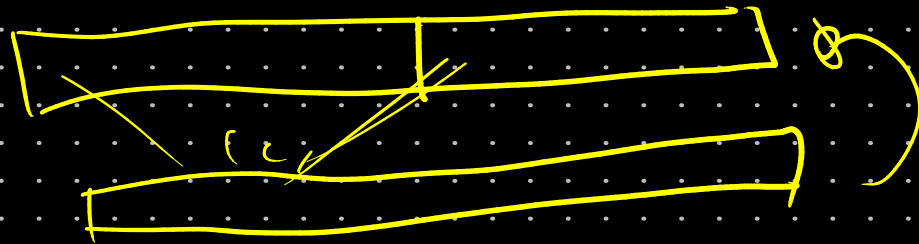
merge sort (L, inicio, inicio +  $\lfloor \frac{fin - inicio}{2} \rfloor$ )

merge sort (L, inicio +  $\lfloor \frac{fin - inicio}{2} \rfloor + 1$ , fin)

merge (inicio, inicio +  $\lfloor \frac{fin - inicio}{2} \rfloor$ , fin, L)  $O(n)$

Lo precisa de una lista auxiliar

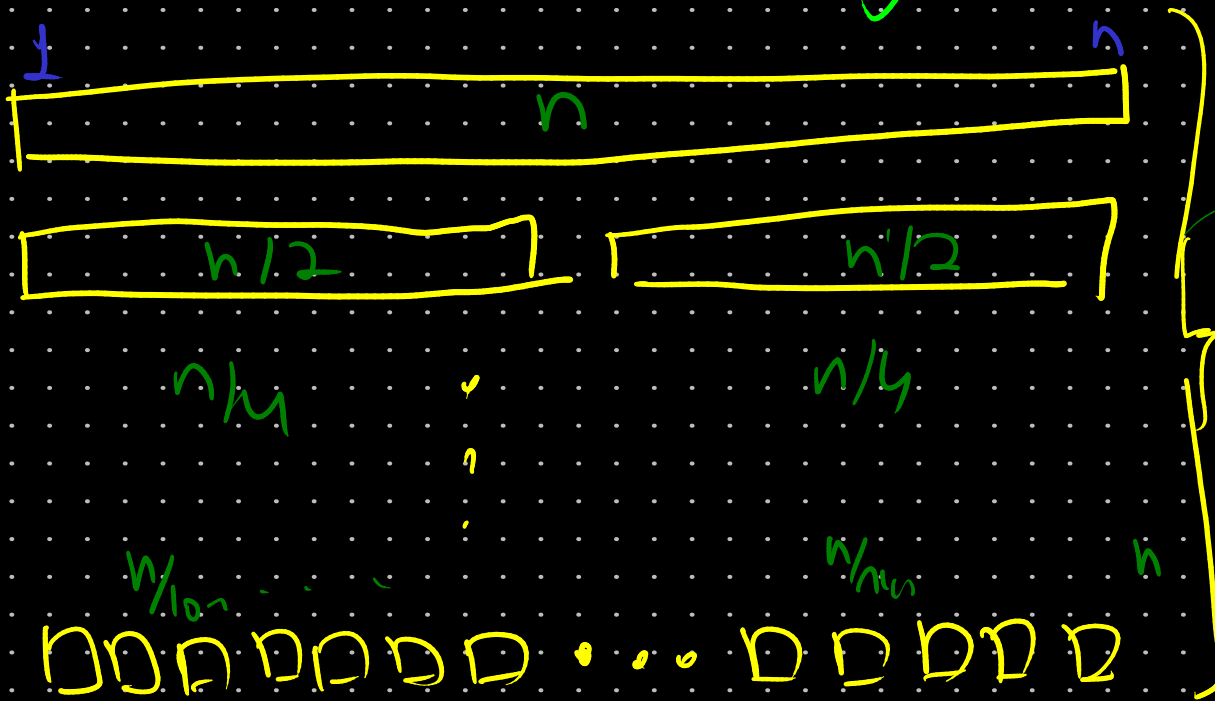
Fin Algoritmo



# Complexidade do Mergesort

Quanto  
vezes  
podemos  
dividir  
um número  
até chegar  
a 1

$\log_2 n$



Quanto  
níveis  
são

$\log(n)$

$O(n \log n)$

# Complexidade do Quicksort

Melhor Caso

pivô igual a mediana



$\leq \log(n)$

$O(\log n)$

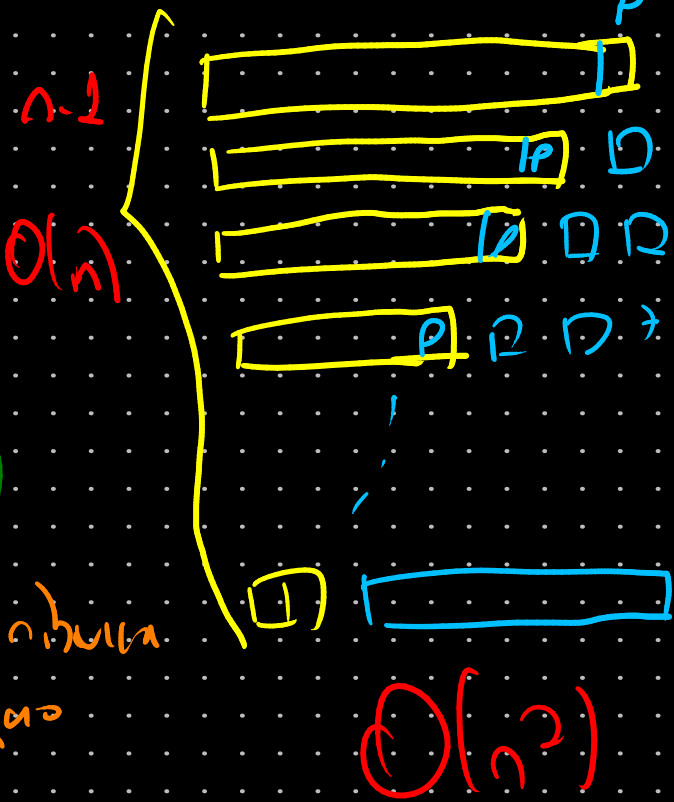
$O(n \log n)$

Depende da distribuição  
na escolha do pivô

Caso Médio  $O(n \log n)$

Pior Caso

pivô extremo



$O(n^2)$

Para minimizar a chance de um pivô ruim,  
Se escolhe como pivô a mediana entre o primeiro  
elemento, o último e o elemento do meio

