

2018/08/09
CC3001

Merge Sort

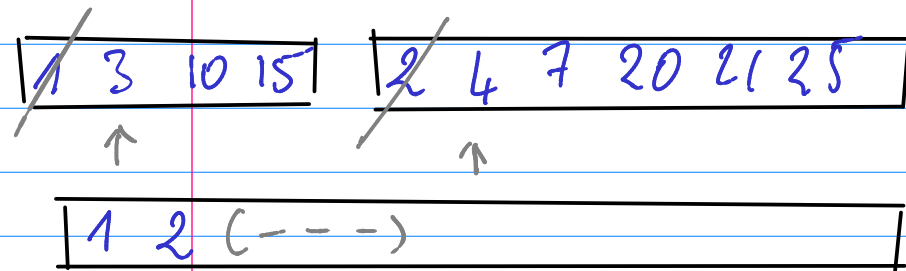
1 3 10 15 2 4 7 20 21 25 76 77

Merge Sort

$O(n \log n)$

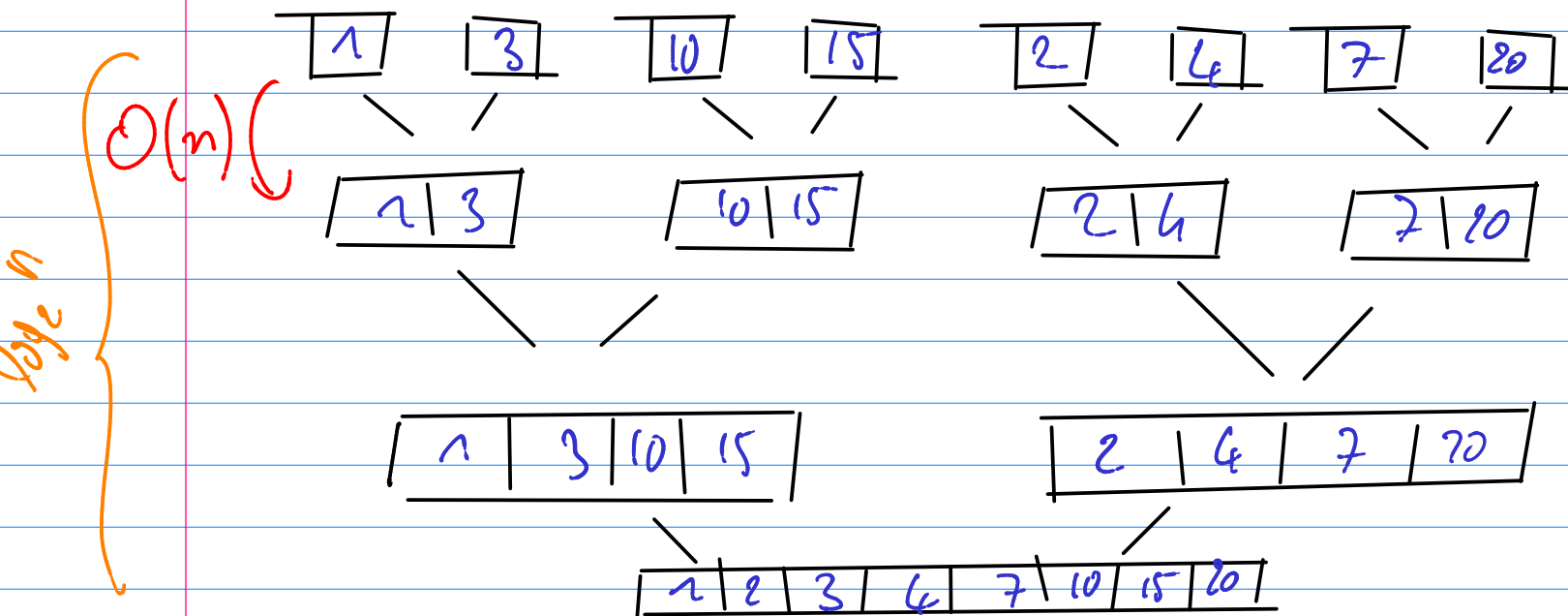
Merge \equiv Mezclar, Fusionar

① Fusionar, "Sorted Merging"



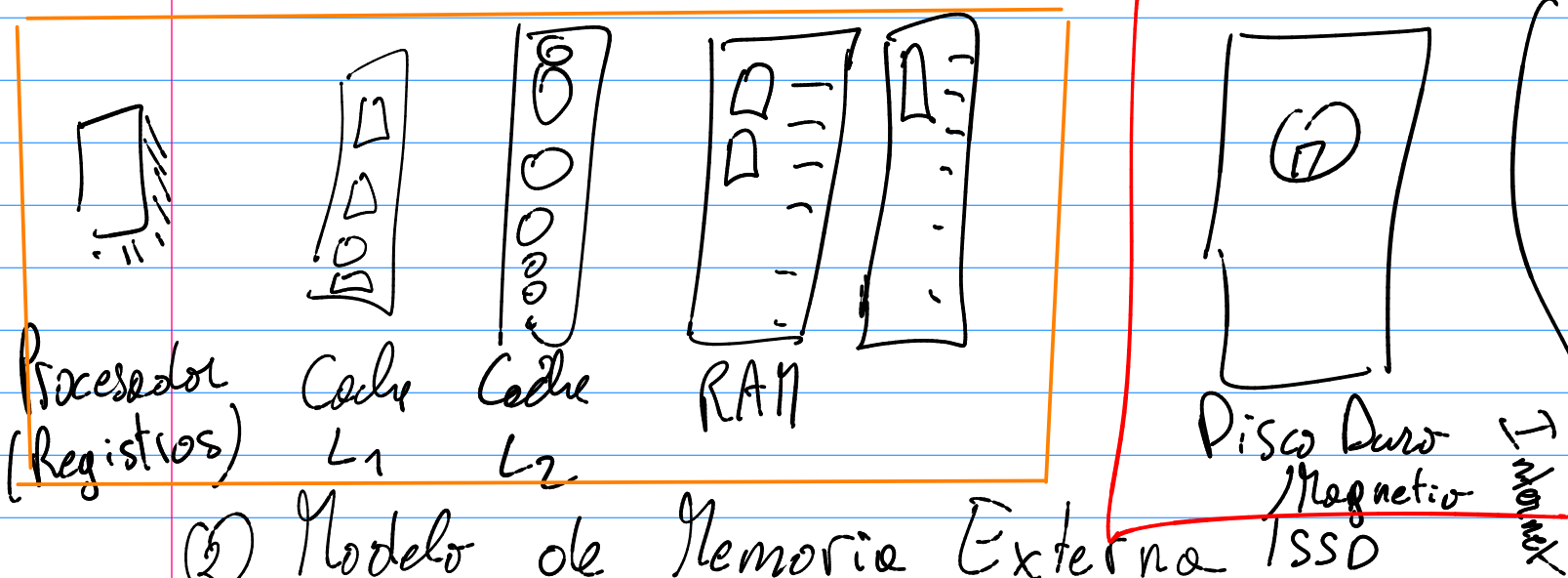
$O(n)$ en el peor caso

② Recursion



Merge Sort | en Memoria externa sobre archivos

1) Hierarquia de Memoria



2) Modelo de Memoria Externa SSD

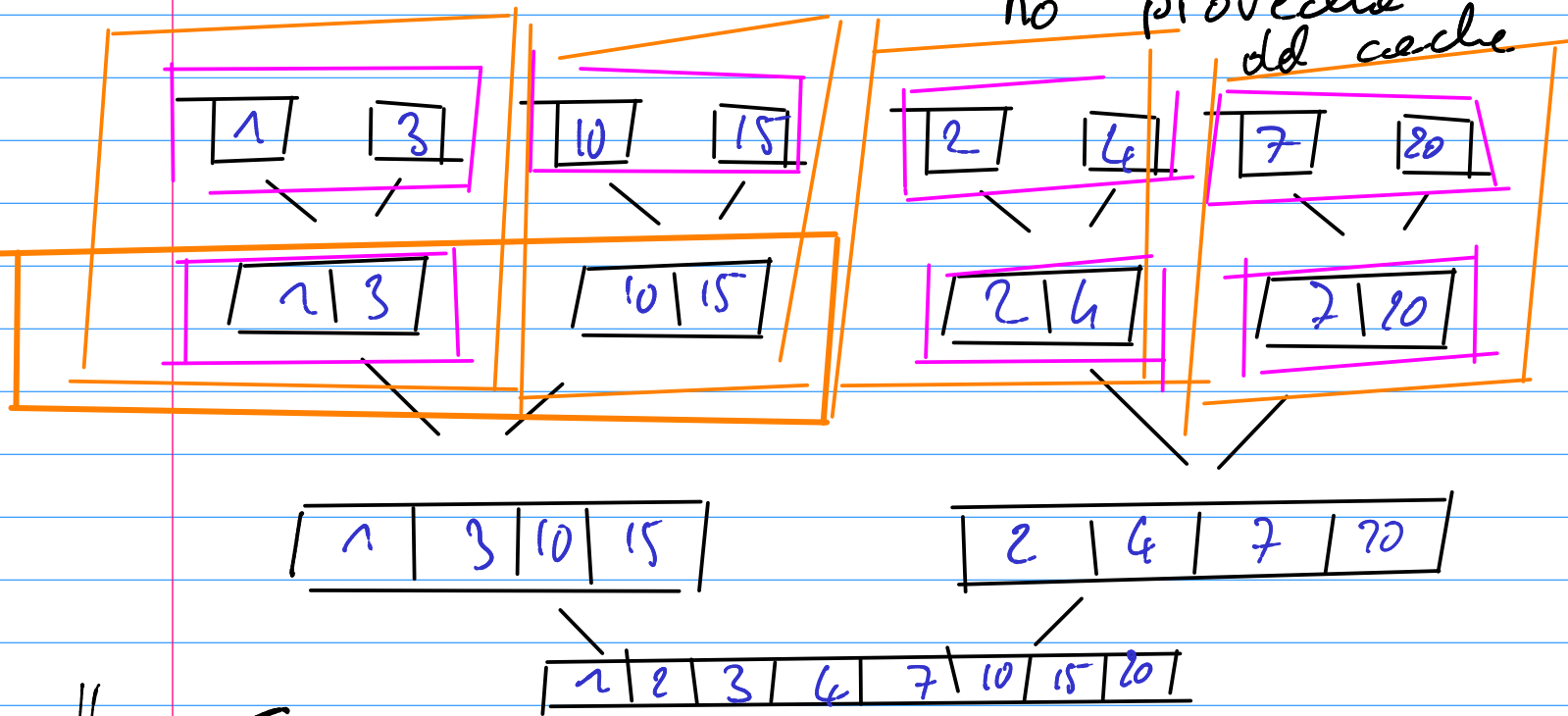
2 niveles de memoria:

- local rapido
- externo lento

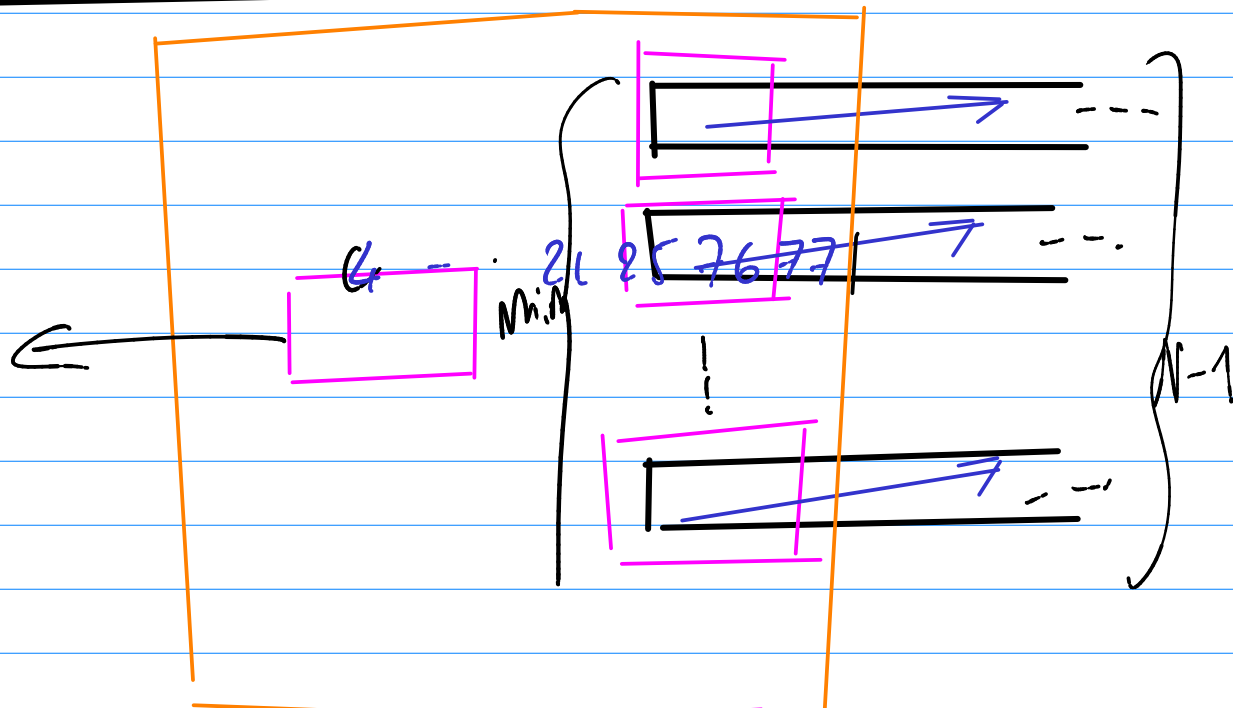
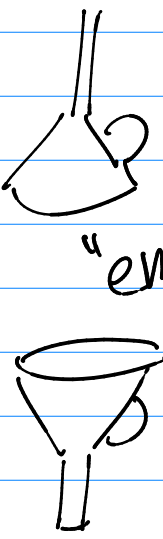
1 tamaño de "paginas" B
1 cache de N paginas

③ Merge Sort en Memoria Externa

$B=2$ $N=2$ \rightarrow El algoritmo original no aprovecha del cache

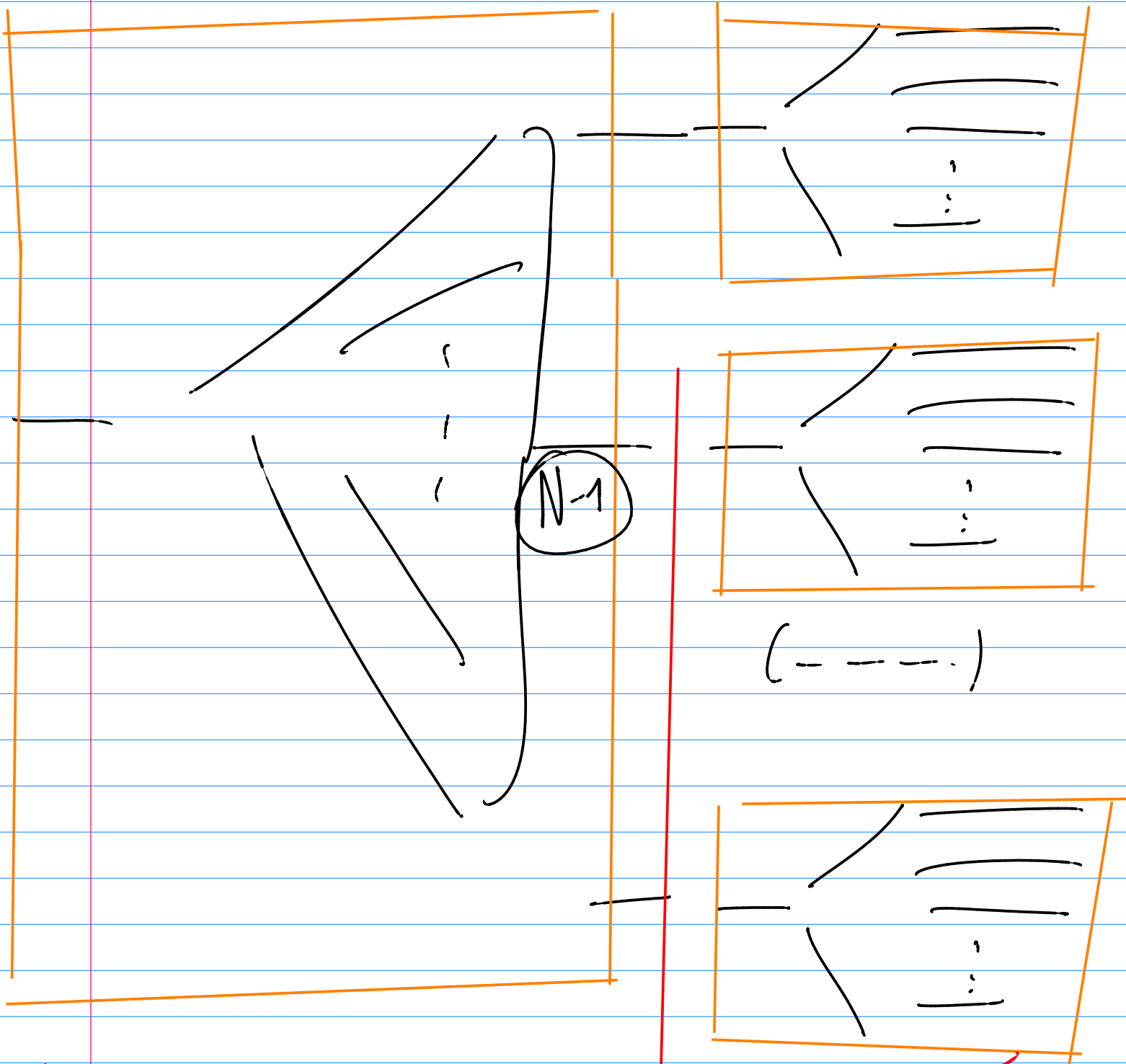


Funnel Sort



n datos $\Rightarrow \frac{n}{B}$ paginas $\Rightarrow \frac{2^N}{B}$ accesos

Analysis de Funnel Sort



$$\frac{\log_2 \frac{n}{B}}{\log_2 (N-1)}$$

$$= \log_{N-1} \frac{n}{B} \text{ fases}$$

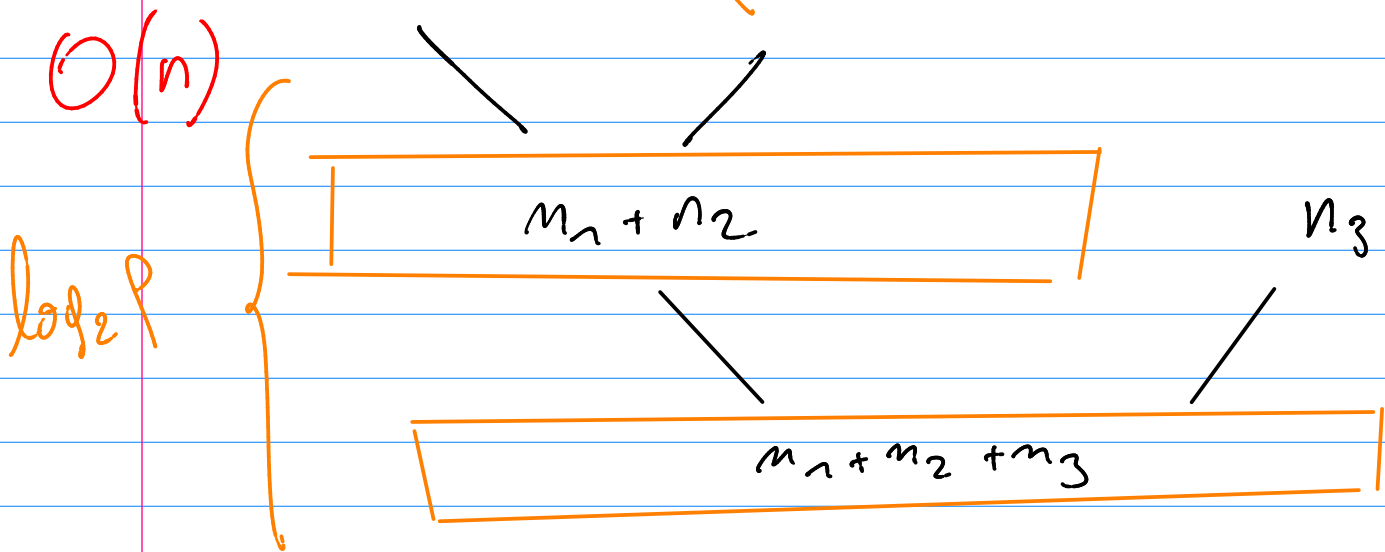
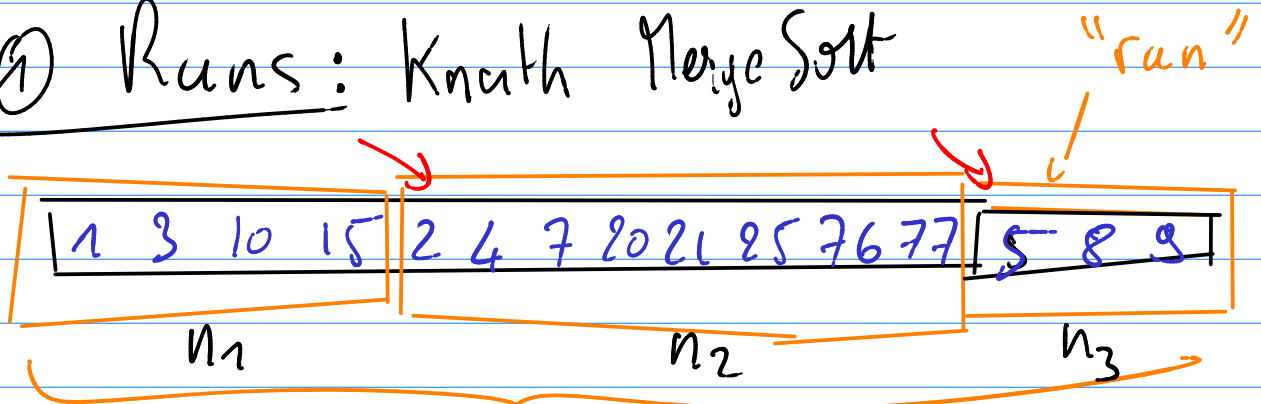
$$O(n/B) \text{ accesos}$$

$$O\left(\frac{n \log(n/B)}{B \log_2(N-1)}\right)$$

accesos

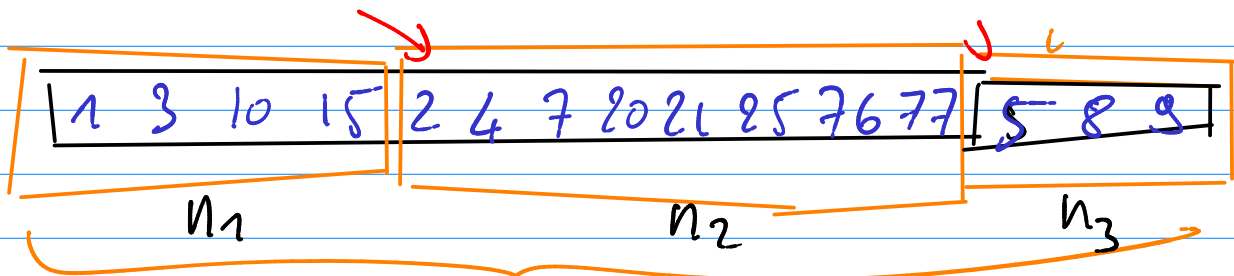
III Adaptive Merge Sort

① Runs: Knuth Merge Sort

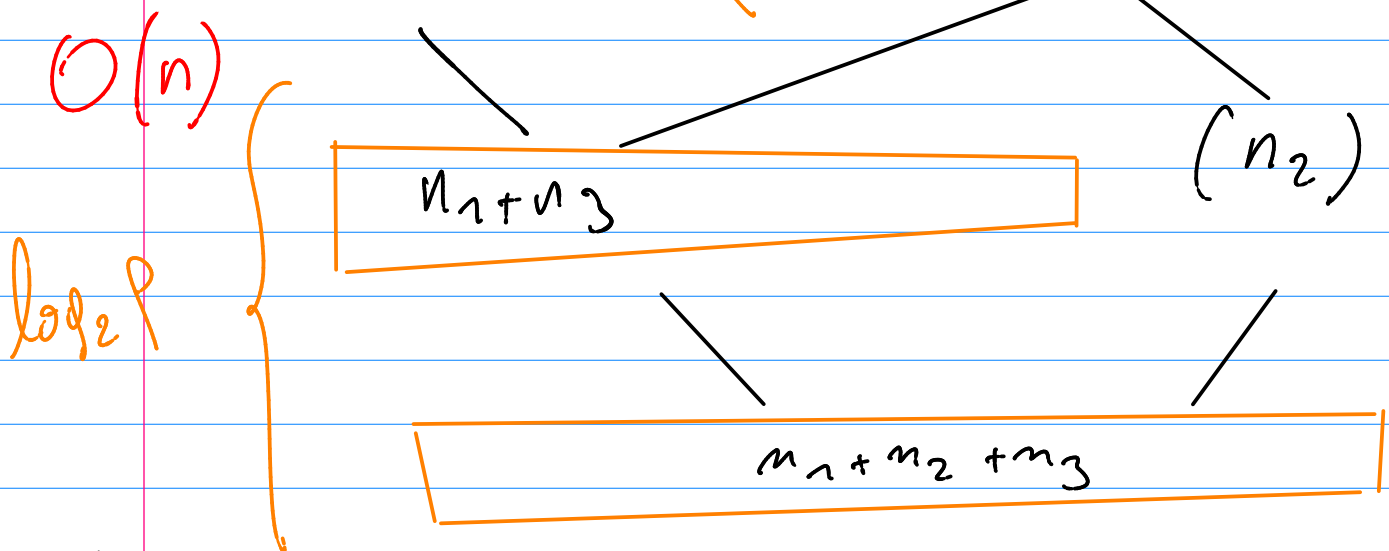


$$O(n(1 + \log_2 \rho))$$

② Entropy Adaptive Merge Sort



$P \equiv \text{"rho"}$



"Arbol de fusión"

- runs mas largos se guardan pero mas tarde
- runs mas cortos se fusionan primeros

Como en Huffman!!!

