

• se i supera $|A| - 1$, finiamo
includendo il però rimanente
di B in C

• se j supera $|B| - 1$, " di A su C .

Questo fase di fusione è lineare ($O(n)$).

Il costo segue la relazione:

$$T(n) \leq \underbrace{2 T(n/2)}_{\text{supera}} + \underbrace{C \cdot n}_{\text{fusione}}$$

Col Master Theorem — e guardando
l'albero ricorrenza — si deduce
quindi che $T(n) = O(n \cdot \log n)$.

→ problema: il merge sort occupa
uno spazio non trascurabile
di memoria per gli
array ausiliari.

→ per grandi quantità di dati, la
non ricorrenza sp. di A è
molto efficiente.

* iniett. $i+j \rightarrow n$ e ad ogni iter.
 i o j scendono.