

Daí, a segunda pergunta pode ser formulada assim

- *Porque a estratégia da divisão acelera o algoritmo da bolha ?*

Quer dizer, na aula 02 a gente viu que isso acontece.

Mas, a gente não perguntou: *porque?*

Só que, agora a gente perguntou.

E agora, a gente pode entender melhor as coisas.

Bom, a gente já sabe que a ordenação da bolha executa em tempo $O(n^2)$.

Daí, porque o tempo é quadrático, ordenar duas metades é mais rápido que ordenar a coisa inteira

$$\frac{n^2}{4} + \frac{n^2}{4} = \frac{n^2}{2} < n^2$$

Mas, a coisa também funcionaria se o algoritmo executasse em tempo $O(n\sqrt{n})$.

Porque

$$\frac{n}{2} \cdot \sqrt{n/2} + \frac{n}{2} \cdot \sqrt{n/2} = \frac{n \cdot \sqrt{n}}{\sqrt{2}} < n\sqrt{n}$$

Pensando desse modo, a gente vê que a estratégia da divisão sempre vale a pena.

A menos que o algoritmo já execute em tempo linear — (*i.e.*, $O(n)$...)

Porque daí

$$\frac{n}{2} + \frac{n}{2} = n$$

Mas, a gente está esquecendo um pequeno detalhe.

Quer dizer, a ordenação das duas metades ainda não produz uma lista completamente ordenada.

Para fazer isso, é preciso realizar a intercalação.

E daí, o tempo de ordenação por metades fica assim

$$\frac{n^2}{4} + \frac{n^2}{4} + n$$

Mas, o ganho quadrático ainda é maior do que o custo linear da intercalação.

Daí que, a coisa vale a pena.

E a coisa também valeria a pena se o algoritmo de base executasse em tempo $O(n\sqrt{n})$.

Veja só,

$$\begin{aligned} \frac{n}{2} \cdot \sqrt{n/2} + \frac{n}{2} \cdot \sqrt{n/2} + n \\ &= \frac{n \cdot \sqrt{n}}{\sqrt{2}} + n \\ &= n \cdot \left(\frac{\sqrt{n}}{\sqrt{2}} + 1 \right) < n\sqrt{n} \end{aligned}$$

Mas, quando o algoritmo de base já executa em tempo $O(n \log n)$, a estratégia da divisão já não é mais efetiva.

Veja só,

$$\begin{aligned} \frac{n}{2} \cdot \log(n/2) + \frac{n}{2} \cdot \log(n/2) + n \\ &= n \cdot \log(n/2) + n \\ &= n \cdot (\log n - \log 2) + n \\ &= n \log n - n + n \\ &= n \log n \end{aligned}$$

Legal, não é?