

divide et impera

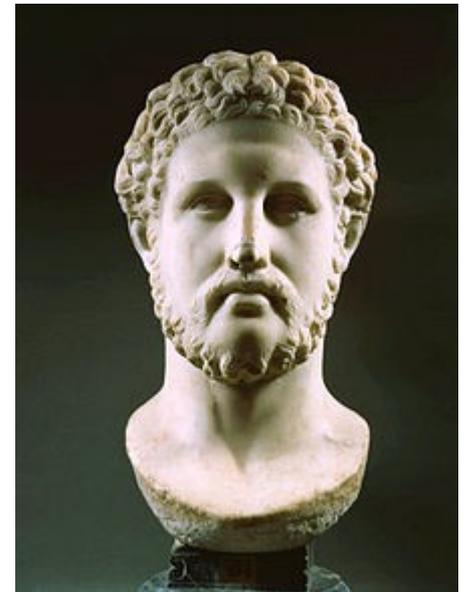


*progetto
CORDA
informatica*

alberto ferrari

origine

- * *Dīvidē et īmpĕrā* «dividi e comanda» è una locuzione latina secondo cui il migliore espediente di una tirannide o di un'autorità qualsiasi per controllare e governare un popolo è dividerlo, provocando rivalità e fomentando discordie
- * La tradizione attribuisce l'origine del motto a Filippo il Macedone (382 a.C. - 336 a.C): διαίρει καὶ βασιλεύει *diàirei kài basíleue*, in greco: «dividi e regna»



- × la tecnica del **divide et impera** è una tecnica di progettazione di algoritmi che si articola in tre fasi:
 - × dividere l'istanza dei dati di ingresso del problema in più sottoistanze disgiunte (*divide*)
 - × risolvere ricorsivamente il problema su ciascuna sottoistanza considerata separatamente
 - × combinare le soluzioni delle sottoistanze per ottenere la soluzione dell'istanza data (*impera*)

efficacia

- × l'efficacia della tecnica dipende dal *metodo di decomposizione* delle istanze e dal *metodo di ricomposizione* delle soluzioni parziali
- × l'efficacia in termini di complessità computazionale dipende dalla capacità di effettuare un *partizionamento bilanciato* dell'istanza dei dati di ingresso
- × la tecnica del divide et impera lavora dall'alto verso il basso
 - × procede dall'istanza generale alle sue sottoistanze

esempi di applicazione (1)

- × algoritmo ricorsivo per calcolare il **massimo** di un **insieme** di n interi
 - × divide l'insieme in due sottoinsiemi disgiunti di $n/2$ interi
 - × calcola ricorsivamente il massimo e di ciascuno dei due sottoinsiemi considerati separatamente
 - × confronta i risultati calcolati per i due sottoinsiemi in modo da determinare il massimo dell'insieme originario

esempi di applicazione (2)

- × algoritmo ricorsivo di ricerca binaria per un array ordinato di n elementi
 - × divide l'array in due sottoarray disgiunti di $n/2$ elementi
 - × poi continua ricorsivamente in uno solo dei due sottoarray

esempi di applicazione (3-4)

- × **mergesort** per un array di n elementi
 - × divide l'array in due sottoarray disgiunti di $n/2$ elementi
 - × ordina ricorsivamente ciascuno dei due sottoarray considerati separatamente
 - × fonde ordinatamente i due sottoarray
- × **quicksort**
 - × divide l'array in tre sottoarray disgiunti composti da elementi contenenti valori \leq o \geq del pivot rispettivamente
 - × ordina ricorsivamente il primo e il terzo sottoarray considerati separatamente
 - × concatena i tre sottoarray
 - × *il fatto che il primo e il terzo sottoarray contengano circa lo stesso numero di elementi dipende in modo critico dalla scelta del pivot*