**DIVIDE ET IMPERA**

***CONDIȚII:***

Metoda se poate aplica în rezolvarea unei probleme care îndeplinește următoarele condiții:

* se poate descompune în (două sau mai multe) subprobleme;
* aceste subprobleme sunt independente una față de alta (subproblema nu se rezolvă pe baza alteia și nu se folosește rezultatul celeilalte);
* aceste subprobleme sunt similare cu problema inițială;
* la rândul lor subproblemele se pot descompune (daca este necesar) în alte subprobleme mai simple;
* aceste subprobleme simple au soluții rapide.

***ETAPE:***

Etapele rezolvării unei probleme inițiale cu Divide et Impera sunt:

* (Împarte) descompunerea problemei inițiale în subprobleme independente, similare problemei de bază, dar de dimensiuni mai mici;
* descompunerea treptată a subproblemelor în alte subprobleme din ce in ce mai simple, până când se pot rezolva imediat, prin algoritmul simplificat;
* (Stăpânește) rezolvarea subproblemelor mici/simple;
* (Combină) combinarea soluțiilor găsite pentru construirea soluțiilor subproblemelor;
* combinarea ultimelor soluții determină obținerea soluției problemei inițiale.

***AVANTAJE:***

* produce algoritmi eficienți
* descompunerea problemei în subprobleme favorizează paralelizarea algoritmului în vederea implementării lui pe mai multe procesoare

***DEZAVANTAJE:***

* se adaugă un overhead datorat recursivității (reținere pe stivă a apelurilor funcțiilor)

***COMPLEXITATEA:***

Complexitatea algoritmilor de tipul Divide et Impera se calculează după formula: T(n) = D(n) + S(n) + C(n)

unde D(n), S(n) ¸si C(n) reprezintă complexitățile celor 3 pași descriși mai sus: împarte (divide), stăpânește (impera) ¸si respectiv combină.

Ex:

Căutare binară- O(log2 n)

Sortarea rapidă (QuickSort) - O(n ∗ log2 n)

Sortarea prin interclasare (MergeSort) - Θ(n ∗ log2 n)

Turnurile din Hanoi - O(2 n )