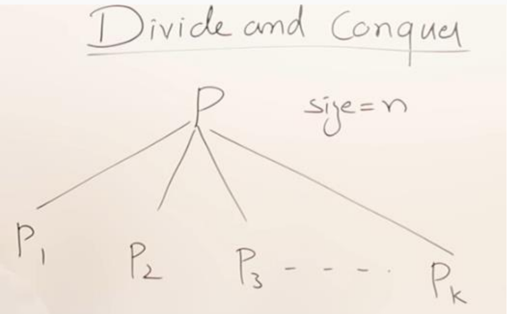
2 Divide And Conquer

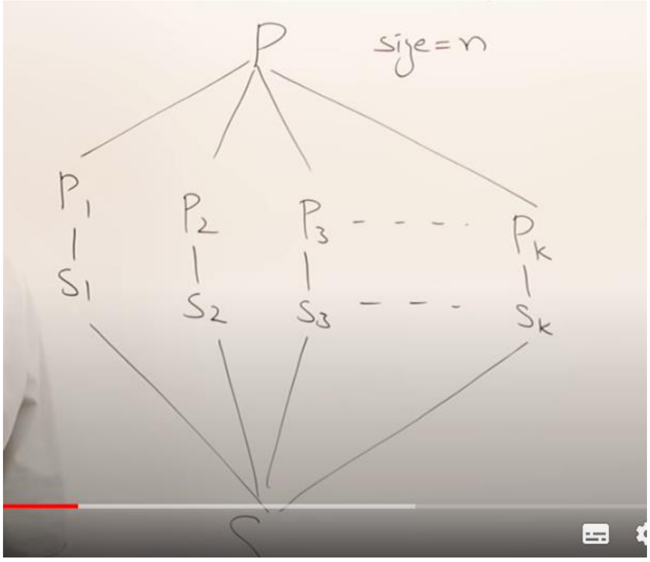


Strategy is approach or design for solving a problem.for solving any computational problem you can adopt this strategy for solving if strategy is suitable for the problem so by practice there is no formula that this is the problem this strategy is applicable though there are a few guidelines so we can follox those guidelines and we can apply the strategy for solving the problem.

We have a problem and size this size iis input for a problem so if a problem p is there of some size and if we say that this problem is large then you can break this problem smaller subproblems P1 P2 P3 goes on as many problems as possible now large problem is broken to smaller subproblems.

So size was larger than we broke that problem to smaller sizes.

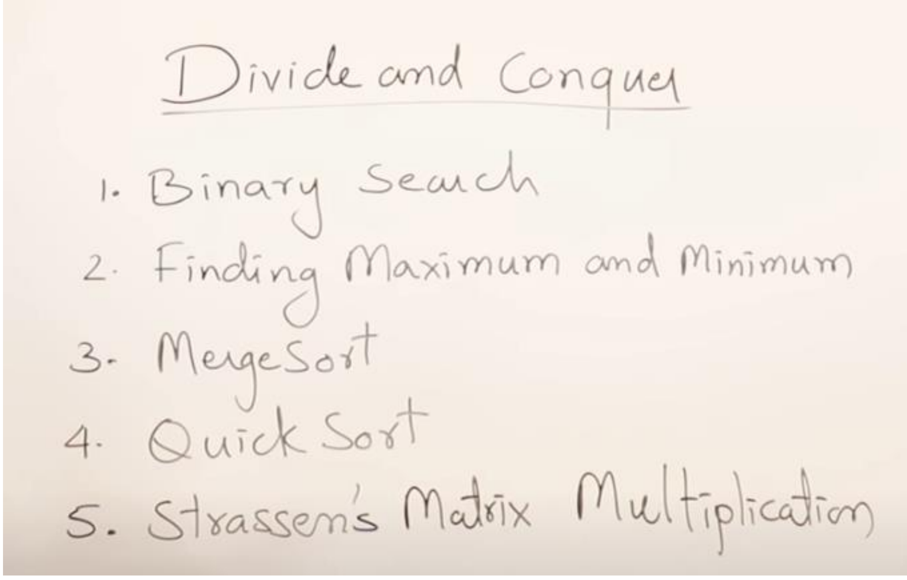
Asagiya devam edelim



Now these subproblems are solved to obtain their solutions all these are individually solved.Once you have to solutions for these then you can combine these solutions to get a solution for main problem.

So your problem is large divide the problem into subproblems and solve those problems and combine the solution of subproblems to get the solution for main problem. It means a problem cannot be solved if it is too big break it into subproblems and solve it. If subproblems is too large then make to same thing apply divide and conquer strategy on subproblems also

One of Important thing for divide and conquer strategy: whatever the problem is the subproblems will be same as main problem. For example if problem is sort then subproblems also be sort .Sunu soyleyelim ornegin bir toplanti yapilacak bunu ufak parcalara ayirdin iste konuklar cagrilcak gibi ama bu divide and conquer degildir.Cunku ayni task capinda olmalidir.



Divide and Conquer

1. Binary Search
2. Finding Maximum and Minimum
3. MergeSort
4. Quick Sort
5. Strassen's Matrix Multiplication

Divide and conquer strategies are recursive. And we should know how to write recursive algorithms how to write functions how to analyze them and how to find time complexities.

So for that we use recurrence relations