Divide and conquer
- general algorithm Design
- Easier to solve smaller instance of a problem then larger one

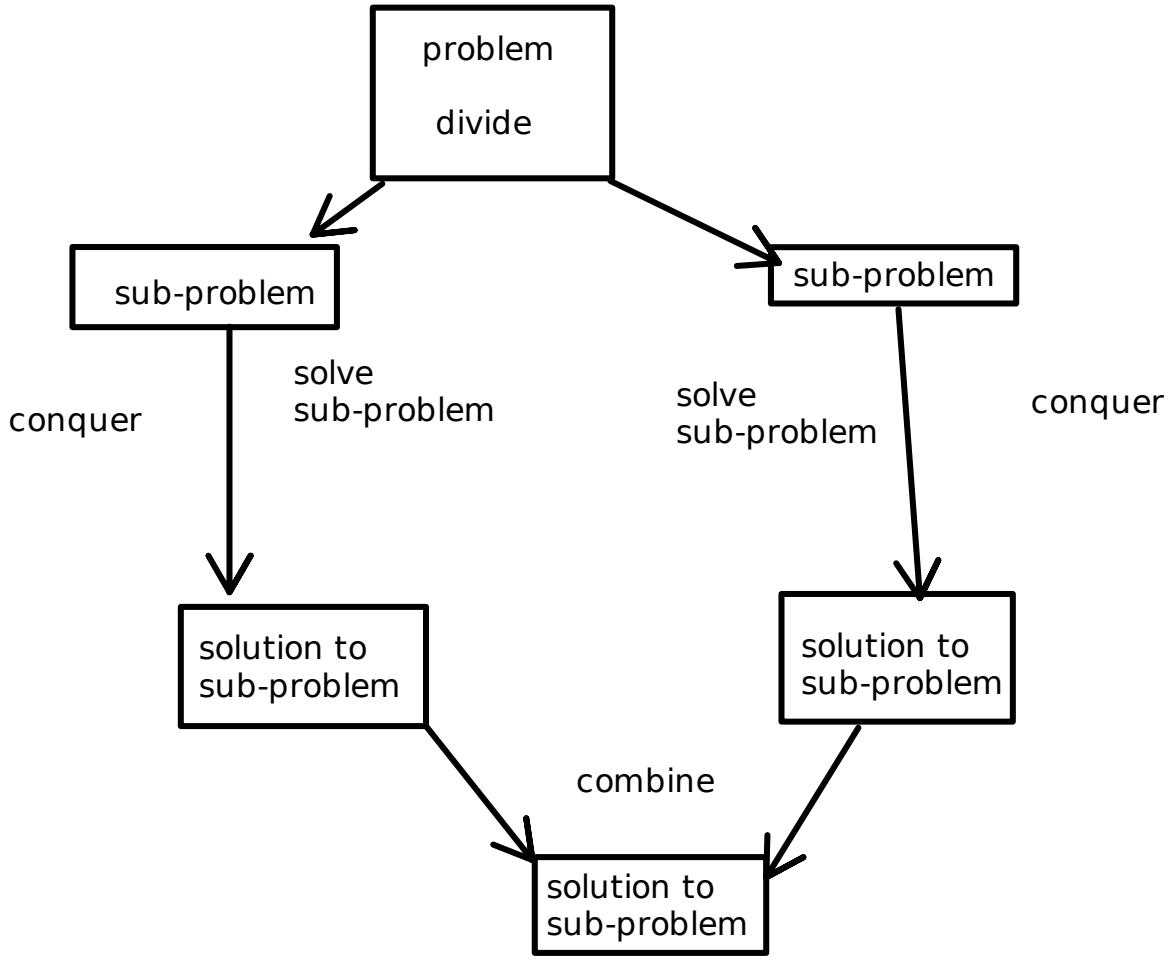

fig: Divide and conquer typical

Solution steps:
1. Divide : devide the problem into a number of smaller subproblems of similar size

2. conquer: The smaller subproblems are solved

3. Combine : The soln obtain are then combined to get the soln of the overall problem.

The general method:

- Given a function to compute on 'n' inputs. The divide-and-conquer strategy suggests splitting the inputs into 'k' distinct subsets, 1<k<=n yeilding 'k' sub-problems.

- These sub-problems must be solved and then a method must be found to combine sub solutions into a solution of the whole.

- If the sub-problems are relatively large. then the divide-and-conquer strategy can possibly be reapplied.

- Often the sub-problems resulting from a divide-and-conquer design are of the same types as original problem

Control Abstraction:

```
Algorithm D and C (p)
{
        if small (p) then return s(p);

        else {
                divide pinto smaller instances p1, p2, ......pk, k>=1
                Apply D and C to each to these subproblems.

                return Combine (D and C(p1), D and C (p2), ..... D and C (pk);
        }

}
```