1) Greedy Algorithm.

→ Mainly useful for optimization problem, optimization problem means maximizing (or) minimizing something.

→ Eg:- Shortest path (weighted or non-weighted).
Longest path (").
Knapsack problem. (fractional)

→ General Structure:-

```
getOptimal(Item arr[], int n)
{
① Initialize res = 0;
② while( All all items are not considered)
  {
    i = SelectAnItem();
    if( feasible(i))
        res = res + i
  }
③ Return res.
}
```

→ Note:- Greedy algorithms doesn't always give you an optimal solution.

* (as Greedy selects the next best not overall best.)

→ Can also be used find optimal close solution for NP hard problems. Eg:- Traveling salesman problem.