

## Greedy Techniques

**Input:** Most of the problems in greedy contain n inputs.

**Output:** Our goal is finding subset which will optimize our objective.

### **Basics:**

**Solution space:** All possible solutions for a given n input all possible solutions(may or may not be correct).

**Feasible solution:** It is one of the solution from solution space which satisfy our condition.

**Optimal solution:** It is one of the feasible solution which optimize our goal.

**Example:** if we have a class of 100 students & we want find the top 10 students

Solution space : all the possible combination(i.e.  $^{100}C_{10}$ )

Feasible solution: that combinations of 10 students which satisfy the objective (i.e. whose avg. marks are greater than specified value)

Optimal solution: that combination of 10 students from feasible solution whose marks are greater than rest other combinations.

### **Control abstraction of greedy:**

```
Greedy_techniques(a,n)
{
  solution=0 // no solution in the beginning
  for(i=1; i,=n; i++)
  {
    X=select(n); // finding the feasible solution
    if(feasible(x)) // checking for feasible soln
      add(x, solution);
  }
}
```

### **Algorithm:**

**Step 1:** Choose an input from the input set, based on some criterion. If no more input exit.

**Step 2:** Check whether the chosen input yields to a feasible solution. If no, discard the input and goto step 1.

**Step 3:** Include the input into the solution vector and update the objective function. Goto step 1.