**Greedy Algorithm**

Greedy algorithm is select only one choice from child node and it does not refer to the choice of the other (another child). And it has two type:

1. Approximate solution: one choice from child and result is approximate from optimal value, this approach using in real life
2. Global optimal solution: one choice from child and result of numbers are correct (equal = optimal value) and not approximate and this approach using in competition

**Example:**

Greedy algorithm is want to get the largest amount so you will choose 10 then 5 and choices made from a short-term perspective (Tactical view) but the best choice is to be 5 then 90 and choices made from a long-term perspective (Strategic view) This algorithm is the disadvantage that it cannot go back to back.

(Do not back down on their options) and that's the main difference between them and Dynamic Programming because they make decisions in view of the previous stages Therefore, this algorithm rethinks after each stage and it is slow to think

**Application:**

* Scheduling: Activity Selection Problem (ASP) - Time, Job
* Graph: Minimum spanning Tree (MST) - Prim’s - Kruskal
* Optimization: Traveling Salesman problem (TSP)

**Greedy Code**

Public class Greedy {

Int swaps, swapf =0;

Public void sort (int[ ] s , int [ ] f) {

For (int i = 0 ; i < s.lenght - 1 ; i++)

For (int j = 0 ; j < s.lenght-1-i ; j++)

If (f[i] > f [j+1])

swapf = f[i]

f[i] = f[j+1]

f[j+1] = swapf

swaps = s[i]

s[i] = s[j+1]

s[j+1] = swaps

}

Public void asp (int[ ]s , int f){

Sort (s , f)

Int i , j =0

For (j = 1 ; j < s.lenght ; j++)

If (s[j]) > f[i])

System.out.println (‘’index No.” + j);

i = j;

}

}