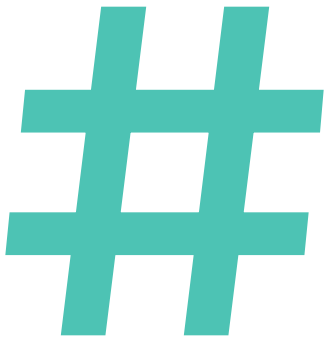

Greedy Algorthim

Step 01 Greedy

A Greedy algorithm is an approach to solving a problem where the solution is fast and is the best solution based on the current situation. Sometimes it will not bring the most optimal result, but it will bring you close to it.

Step 02 Class Attendance Problem

Consider you have the list of classes below, and you want to attend the most classes you can

Math 9-10

CS 9:30-10:30

English 10-11

Programming 10:30-11:30

History 11-12

02 Solution

1. Pick the class that ends the soonest. This is the first class you'll hold in this classroom.

2. Now, you have to pick a class that starts after the first class. Again, pick the class that ends the soonest. This is the second class you'll hold.

03 Result

Math 9-10

English 10-11

History 11-12

The greedy algorithm is working very well in solving the problem and it does bring the most optimal solution to the class attendance problem.

Step 03 Example

Travel plan problem

Consider you have to travel to Europe and you have a list of places you want to visit during your trip. You have a time capacity of 3 hours only.

You decided to list your places with the time each place will take and a score of points representing how much you want to visit the place.

We will use a Greedy algorithm to solve this problem and create a plan containing the places you should visit.

place 1

time: 1 hour

score: 5 points

place 2

time: 2 hours

score: 10 points

place 3:

time: 1 hour

score: 15 points

place 4

time: 3 hour

score: 5 point

01 Solution

- Pick the place that has the most points and will fit into the time capacity you have.
- Pick the next place that has the most points and will fit into the remaining time capacity you have.

And so on.

02 Result

Plan
Name: Place 3 Time: 1 Score: 15
Name: Place 2 Time: 2 Score: 10

03 Implementation

```
class Place {

    String name;
    int time;
    int score;

    Place(String name, int time, int score) {
        this.name = name;
        this.time = time;
        this.score = score;
    }

}

public class TravelPlan {

    int timeCapacity = 3;
    Place places[] = { new Place
(“Place 1”, 1, 5), new Place(“Place 2”, 2, 10), new Place(“Place 3”, 1,15),
            new Place(“Place 4”, 3, 5) };


    public Place[] greedyAlgorithm() {

        Place plan[] = new Place[this.places.length];
        Place[] temp_place_arr = this.places;
        Place currentPlace = temp_place_arr[0];
        int currentTimeCapacity = 0;

        // counter for plan []
        int k = 0;
        // index of the element to delete
        int removeIndex = 0;

        while (currentTimeCapacity < this.timeCapacity) {

            // find highest score place that fits the time capacity
            for (int i = 0; i < temp_place_arr.length; i++) {

                if (temp_place_arr[i].time <=
                   (this.timeCapacity - currentTimeCapacity)
                        && temp_place_arr[i].score >= currentPlace.score) {
                    currentPlace = temp_place_arr[i];
                    removeIndex = i;

                }

            }

```

```
            // add the place to the plan
            plan[k++] = currentPlace;
            // add the place time to the time capacity counter
            currentTimeCapacity += currentPlace.time;
            // delete the place from the temp array
            delete(temp_place_arr, removeIndex);
            currentPlace = temp_place_arr[0];
        }

        return plan;
    }

    public Place[] delete(Place[] temp_place_arr, int elementIndex) {

        for (int i = elementIndex; i < temp_place_arr.length - 1; i++) {
            temp_place_arr[i] = temp_place_arr[i + 1];
        }

        return temp_place_arr;

    }

    public void display() {
        for (Place place : places) {
            System.out.println
(“Name: “ + place.name + “ Time: “ + place.time + “ Score: “ + placescore);

        }
    }

    public void displayPlan(Place plan[]) {

        for (Place place : plan)
            if (place != null)
                System.out.println
(“Name: “ + place.name + “ Time: “ + place.time + “ Score: “ + placescore);

    }

    public static void main(String[] args) {
        TravelPlan tp = new TravelPlan();

        System.out.println();
        System.out.println(“List of places : “);
        tp.display();

        System.out.println();

        Place[] plan = tp.greedyAlgorithm();
        System.out.println(“Plan : “);
         tp.displayPlan(plan);
```

```
101             System.out.println();
102
103         }
104
105     }
```

OUTPUT

```
List of places :
Name: Place 1 Time: 1 Score: 5
Name: Place 2 Time: 2 Score: 10
Name: Place 3 Time: 1 Score: 15
Name: Place 4 Time: 3 Score: 5

Plan :
Name: Place 3 Time: 1 Score: 15

Name: Place 2 Time: 2 Score: 10
```