

**Module Name** - Dynamic Programming

**Topic Name:** Introduction to Greedy Algorithm

# Time Allocation Summary

Element	Slide numbers	Maximum time(min)
Today's Agenda	5	10
Introduction to Greedy Algorithm	3-7	60
Advantages & Disadvantages of Greedy Algorithm	8-8	20
When to Use Greedy	9-11	30
<b>Total Time</b>		

# Today's Agenda

- Introduction to Greedy Algorithms
- Advantages & Disadvantages of Greedy Algorithm
- Greedy Algorithms (When to Use)

In the field of computer science, there is never a single solution for every type of problem.

Here we will learn about Greedy Algorithm which is used to solve optimization problems.

Optimization problems are those where you have to find the best solution out of all the possible solutions.

Let us understand the “Greedy approach” to solve problems in detail.

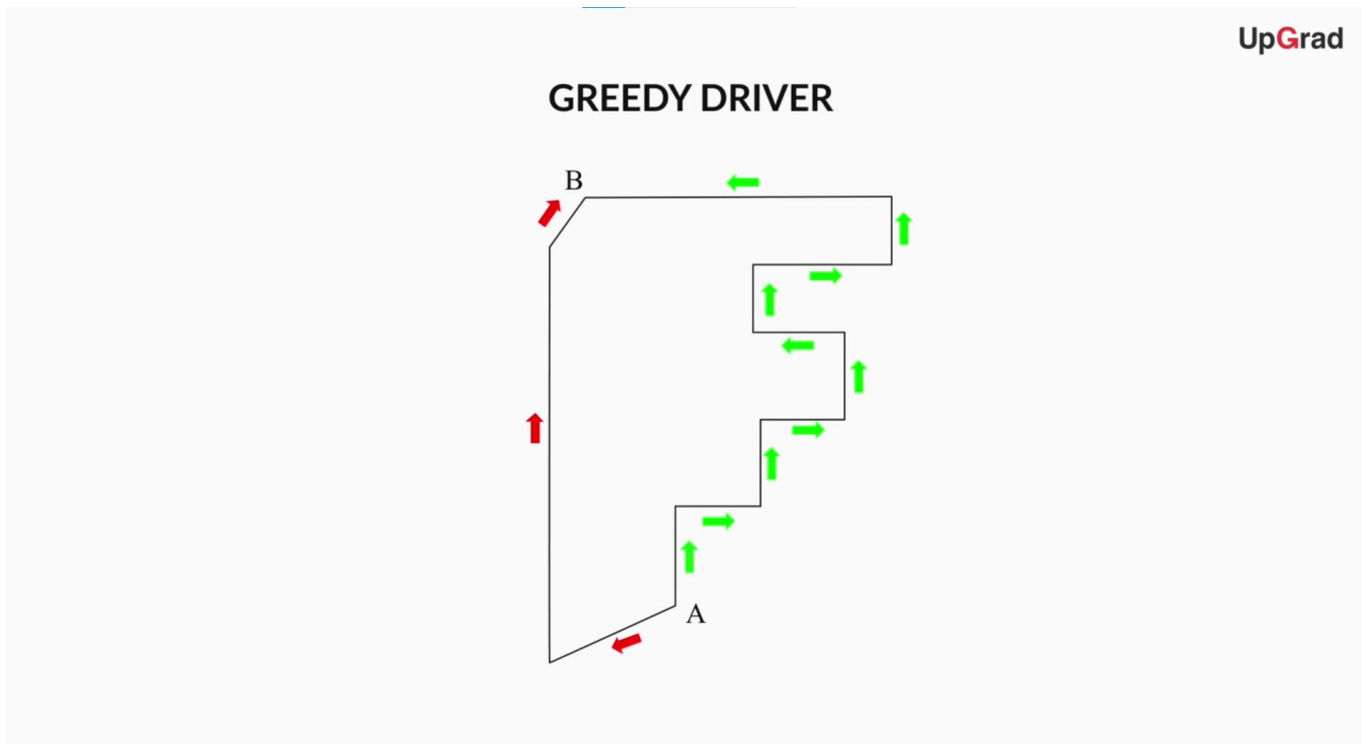
As we already know, Greedy algorithms are used to solve optimization problems.

The idea is to make locally optimum choice with the hope that it would also be the globally optimum solution.

It is important to know that while following the greedy approach, we take decisions which are good for **short term**.

Also there is a high chance that the approach may not work for the given problem, i.e. not give the optimal solution. Therefore when it works, we usually have to provide a proof of its correctness.

In order to understand the proof of its feasibility let us go through an example.



If we use the greedy approach here, it would choose the path which would seem most optimal at that time without evaluating other options.

Evidently here the approach would not work.

Let us understand the advantages and disadvantages of greedy algorithms.

## ADVANTAGES AND DISADVANTAGES

### Advantages

1. Easy to come up with
2. Easy to describe and implement → Easy to analyze for Time and Space Complexity

### Disadvantages

1. Tough to find the right approach – might not lead to globally optimal result
2. Difficult to prove correctness – requires nuanced proof



- Greedy is an algorithmic paradigm that builds up a solution piece by piece, always choosing the next piece that offers the most obvious and immediate benefit.
- So the problems where choosing locally optimal also leads to global solution are best fit for Greedy.
- An optimization problem can be solved using Greedy if the problem has the following property:
  - *At every step, we can make a choice that looks best at the moment, and we get the optimal solution of the complete problem.*

- If a Greedy Algorithm can solve a problem, then it generally becomes the best method to solve that problem as the Greedy algorithms are in general more efficient than other techniques like Dynamic Programming. **But Greedy algorithms cannot always be applied.**
- For example, the Fractional Knapsack problem can be solved using Greedy, but 0-1 Knapsack cannot be solved using Greedy.



Thank You!