

Greedy Algorithm

Omer Danish (1505053)
Sagor Biswas (1505060)



Department of Computer Science and Engineering
Bangladesh University of Engineering and Technology
(BUET)

Dhaka 1000
October 5, 2020

Contents

1	Introduction	3
2	Advantages of Greedy Algorithm	3
3	Example	3
4	Problems with Greedy Algorithm	4
5	Where and When to use	6
6	Applications of Greedy Algorithm	6
7	Conclusion	6

1 Introduction

A greedy algorithm is an algorithmic paradigm that follows the problem solving heuristic of making the locally optimal choice at each stage with the intent of finding a global optimum.

In another words, A greedy algorithm constructs an object X one step at a time, at each step choosing the locally best option. Then constructs the globally best object by repeatedly choosing the locally best option.

Greedy Algorithm is simple, easy to implement and runs fast But very often they don't provide a globally optimum.

2 Advantages of Greedy Algorithm

A greedy algorithm is a mathematical process that looks for simple, easy-to-implement solutions to complex, multi-step problems by deciding which next step will provide the most obvious benefit.

Simplicity: Greedy algorithms are often easier to describe and code up than other algorithms.

Efficiency: Greedy algorithms can often be implemented more efficiently than other algorithms.

3 Example

Now, we shall look at an example.

Suppose, there is a very rich man who has infinite supply of 500 Taka ,100 Taka and 50 Taka notes. He has a friend who needs some money. He asks for 650 Taka but in minimum number of notes.

If there was no restriction, he could have given it in many ways. In any combination of 500, 100 and 50 taka that sums 650.

But they don't satisfy the condition of minimum note number.

We can use Greedy approach to solve this problem.

First, we shall find the biggest note less than 650, which is 500 Taka. So, the sum is 500.

Then, we shall find the biggest note less than $650 - 500 = 150$, which is 100 Taka. Now the sum is 600.

Finally we shall choose the note which is less or equals to $650 - 600 = 50$ Taka. And that is our answer. One 500 Taka, One 100 Taka, One 50 Taka.

For every iteration we chose note of largest value that does not take us past the amount to be given.
This algorithm is Greedy Algorithm.

4 Problems with Greedy Algorithm

Greedy algorithms have several drawbacks:

Hard to design: Once you have found the right greedy approach, designing greedy algorithms can be easy. However, finding the right approach can be hard.

Hard to verify: Showing a greedy algorithm is correct often requires a nuanced argument.

Now, we shall look at an example where greedy algorithm will provide us wrong answer.

We know from greedy algorithm that if we choose local optimal at each stage it will form a global optimum. It is true in some cases but not for all time.

Let's have a look at the figure above .

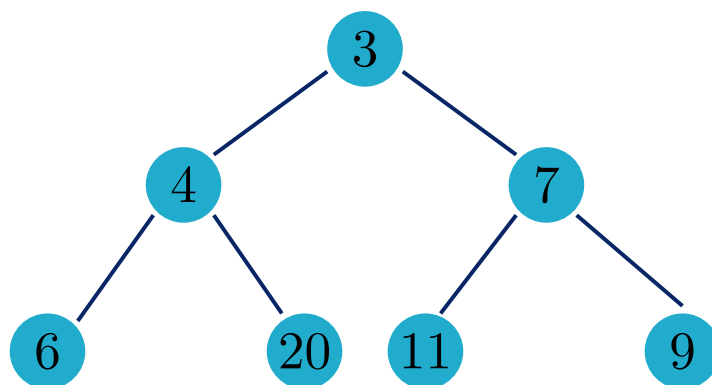


Figure 1: Graph

We want to find the maximum value of addition from root node to leaf node.

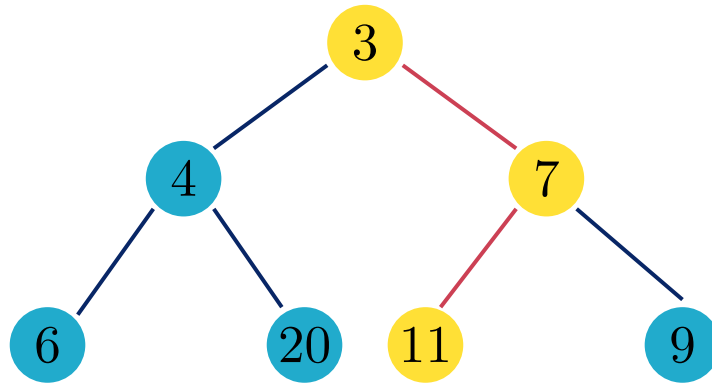


Figure 2: Found path with Greedy Algorithm

If we approach according to greedy algorithm we get a value that is $3+7+11 = 20$, which is not maximum value of addition.

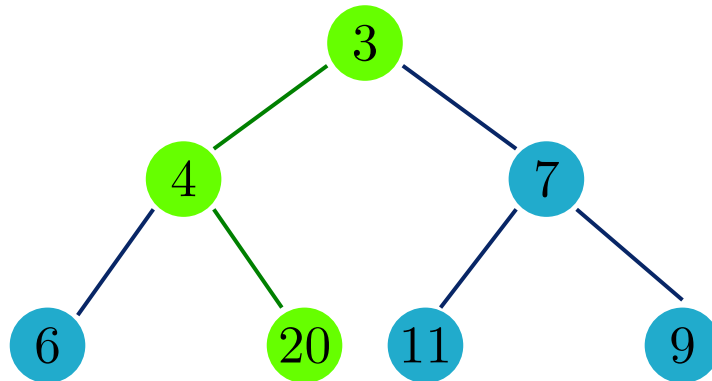


Figure 3: Actual Solution to the problem

Where the actual optimal solution is $3+4+20 = 27$

That means in this example greedy algorithm is unable to find global optimum.

5 Where and When to use

Problems on which greedy approach work has two properties

Greedy-choice property Once you have found the right greedy approach, designing greedy algorithms can be easy. However, finding the right approach can be hard.

Optimum substructure property An optimum solution to the problem contains an optimum solution to

6 Applications of Greedy Algorithm

- Huffman Coding
- Fractional Knapsack Problem
- Prim's Minimum Spanning Tree
- Job Sequencing Problem
- Activity Selection Problem

7 Conclusion

- Greedy algorithms aim for global optimality by iteratively making a locally optimal decision.
- To show correctness, typically need to show
 - The algorithm produces a legal answer, and
 - The algorithm produces an optimal answer.
- Often use “greedy stays ahead” to show optimality