# GeeksforGeeks

A computer science portal for geeks

GeeksQuiz

- Home
- Algorithms
- DS
- GATE
- Interview Corner
- Q&A
- C
- C++
- Java
- Books
- Contribute
- Ask a Q
- About

Array
Bit Magic
C/C++
Articles
GFacts
Linked List
MCQ
Misc
Output
String
Tree
Graph


# Dynamic Programming | Set 2 (Optimal Substructure Property)

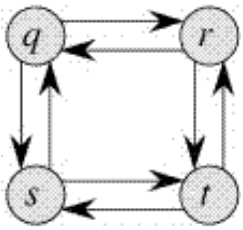
As we discussed in Set 1, following are the two main properties of a problem that suggest that the given problem can be solved using Dynamic programming.

1) Overlapping Subproblems
2) Optimal Substructure

We have already discussed Overlapping Subproblem property in the Set 1. Let us discuss Optimal Substructure property here.

**2) Optimal Substructure:** A given problems has Optimal Substructure Property if optimal solution of the given problem can be obtained by using optimal solutions of its subproblems.
For example the shortest path problem has following optimal substructure property: If a node x lies in the

shortest path from a source node u to destination node v then the shortest path from u to v is combination of shortest path from u to x and shortest path from x to v. The standard All Pair Shortest Path algorithms like Floyd–Warshall and Bellman–Ford are typical examples of Dynamic Programming.
On the other hand the Longest path problem doesn't have the Optimal Substructure property. Here by Longest Path we mean longest simple path (path without cycle) between two nodes. Consider the following unweighted graph given in the CLRS book. There are two longest paths from q to t: q -> r ->t and q ->s->t. Unlike shortest paths, these longest paths do not have the optimal substructure property. For example, the longest path q->r->t is not a combination of longest path from q to r and longest path from r to t, because the longest path from q to r is q->s->t->r.




We will be covering some example problems in future posts on Dynamic Programming.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.

**References:**
http://en.wikipedia.org/wiki/Optimal_substructure
CLRS book

## Related Topics:

- Linearity of Expectation
- Iterative Tower of Hanoi
- Count possible ways to construct buildings
- Build Lowest Number by Removing n digits from a given number
- Set Cover Problem | Set 1 (Greedy Approximate Algorithm)
- Find number of days between two given dates
- How to print maximum number of A's using given four keys
- Write an iterative O(Log y) function for pow(x, y)

Tags: Dynamic Programming

Tweet 0 +1 0

**Writing code in comment?** Please use **ideone.com** and share the link here.

12 Comments GeeksforGeeks 1 Login
Recommend Share Sort by Newest
Join the discussion…

**shiwakant.bharti** • 2 years ago

Example from wiki where the substructure may not be optimal.
http://en.wikipedia.org/wiki/O...

Least-cost airline fare. (Using on online flight search, we will frequently find that the cheapest flight from airport A to airport B involves a single connection through airport C, but the cheapest flight from airport A to airport C involves a connection through some other airport D.)

3 • Reply • Share ›

**SDK** • 4 years ago

Please can somebody clarify the difference between Greedy and Dp..solutions..

i.e how to decide which technique to use when by providing examples. I think that would be nice post :)
Thank u

• Reply • Share ›

**Anand** • 4 years ago

Here is blog that has all solved DP problem frequently asked in interviews.

http://anandtechblog.blogspot.com/2011/01/amazon-question-dynamic-programming.html

1 • Reply • Share ›

**tk** • 4 years ago

As far as I know, most of the optimization problems have optimal substructure property. It is mainly Overlapping subproblem property that helps us in deciding to choose DP. Does anyone know any example optimization problem - other than the longest path - that doesn't have the optimal substructure property?

• Reply • Share ›

**shiwakant.bharti** ➔ tk • 2 years ago

Example from wiki where the substructure may not be optimal.
http://en.wikipedia.org/wiki/O...

Least-cost airline fare. (Using on online flight search, we will frequently find that the cheapest flight from airport A to airport B involves a single connection through airport C, but the cheapest flight from airport A to airport C involves a connection through some other airport D.)

• Reply • Share ›

**pira** ➔ shiwakant.bharti • 10 months ago

i didnt get the above example.
can anyone explain it ?

**aj12009_3** ➚ pira • 8 months ago

lets say you wanna go from A to B, website will show you the cheapest rates possible via a connection from point C. But if you think that sum is A to C(Cheapest) + C to B(Cheapest)[It might be costlier than the website is offering]. that's would not be the case. As cheapest from A to C is via a point D.

Thus you can't find cheapest possible fares from A to B , by finding cheapest fares of flights via connecting airports for that route.

1 ▲ | ▼ • Reply • Share ›

**Vineel** ➚ pira • 9 months ago

Go through the above article, to find the longest path from vertex to another is an optimization problem, but it does not have a substructure. i.e, longest path from A to B can not be viewed as longest path from A to C and from C to A.

▲ | ▼ • Reply • Share ›

**Venki** • 4 years ago

The "Optimal Substructure Property" also called as "principle of optimality". In order to apply DP, solution to the problem should satisfy principle of optimality. However, determining the Principle of Optimality is little difficult task, hope some of the upcoming examples will clarify.

For example finding optimal solution to one sub-instance may prevent choosing optimal solution to another instance, i.e. the optimal instances are not independent.

▲ | ▼ • Reply • Share ›

**rocky** • 4 years ago

Nice Post! What about the Dijekstra Algorithm. Dijekstra also follow optimal substructure property. Isn't it a DP algo?

1 ▲ | ▼ • Reply • Share ›

**Jagat** ➚ rocky • 2 years ago

In case of Disjkstra, you evaluate a specific decision that moves you directly towards the optimal solution, and that is the property of a greedy algorithm.

On the other hand, when using DP, you've no idea what the optimal solution is till you've evaluated the optimal solution to all the possible sub problems.

8 ▲ | ▼ • Reply • Share ›

**Shiraj Pokharel** ➚ rocky • 4 years ago

No its greedy my dear.

1 ▲ | ▼ • Reply • Share ›

Subscribe Add Disqus to your site Privacy


- Interview Experiences
- Advanced Data Structures
- Dynamic Programming
- Greedy Algorithms
- Backtracking
- Pattern Searching
- Divide & Conquer
- Mathematical Algorithms
- Recursion
- Geometric Algorithms

## Popular Posts

- All permutations of a given string
- Memory Layout of C Programs
- Understanding “extern” keyword in C
- Median of two sorted arrays
- Tree traversal without recursion and without stack!
- Structure Member Alignment, Padding and Data Packing
- Intersection point of two Linked Lists
- Lowest Common Ancestor in a BST.

- Check if a binary tree is BST or not
- Sorted Linked List to Balanced BST

- Follow @GeeksforGeeks

# Recent Comments

- lt_k

  i need help for coding this function in java...

  Java Programming Language · 1 hour ago

- Piyush

  What is the purpose of else if (recStack[*i])...

  Detect Cycle in a Directed Graph · 1 hour ago

- Andy Toh

  My compile-time solution, which agrees with the...

  Dynamic Programming | Set 16 (Floyd Warshall Algorithm) · 1 hour ago

- lucy

  because we first fill zero in first col and...

  Dynamic Programming | Set 29 (Longest Common Substring) · 2 hours ago

- lucy

  @GeeksforGeeks i don't n know what is this long...

  Dynamic Programming | Set 28 (Minimum insertions to form a palindrome) · 3 hours ago

- manish

  Because TAN is not a subsequence of RANT. ANT...

  Given two strings, find if first string is a subsequence of second · 3 hours ago

@geeksforgeeks, Some rights reserved Contact Us!
Powered by WordPress & MooTools, customized by geeksforgeeks team