Dynamic Programming

Suvasish Das

Introduction

Dynamic Programming[1] (DP) is mainly an optimization over plain recursion. Problems that have an option to choose from can be solved using DP. The idea is to store the results of subproblems so that we do not have to re-compute them when needed later. This simple optimization reduces time complexities from exponential to polynomial.

To dynamically solve a problem, we need to check two necessary conditions:

1. Overlapping Subproblems: When the solutions to the same subproblems are needed repetitively for solving the actual problem. The problem is said to have overlapping subproblems property.

2. Optimal Substructure Property: If the optimal solution of the given problem can be obtained by using optimal solutions of its subproblems then the problem is said to have Optimal Substructure Property.

Overlapping Subproblem

A problem is said to have overlapping subproblems[2] if the problem can be broken down into subproblems which are reused several times OR a recursive algorithm for the problem solves the same subproblem over and over rather than always generating new subproblems.

A second ingredient that an optimization problem must have for dynamic programming to apply is that the space of subproblems must be ”small” in the sense that a recursive algorithm for the problem solves the same subproblems over and over, rather than always generating new subproblems (Introduction to Algorithms by CLRS)

Optimal Substructure

1 Longest Palindromic Substring

Problem Statement

Given a string s, return the longest palindromic substring in s.

```
Example1: I/P = "babad" O/P = "bab"
Example2: I/P = "cbbd" O/P = "bb"
```

Brute Force Solution

List all substring and check if it is palindromic or not. Intuitively, to check wheather a string is palindromic or not, takes $O(n)$ time complexity. And, we are iterating through the string in total $O(n^2)$ time. So, the time-complexity of the solution will be $O(n^3)$

Using Recursion

Using Dynamic Programming

Using Sliding Window

References

[1] "How to solve a Dynamic Programming Problem ?," *GeeksforGeeks*, Apr. 12, 2017. https://www.geeksforgeeks.org/solve-dynamic-programming-problem/

[2] "What are overlapping subproblems in Dynamic Programming (DP)?," *Stack Overflow.* https://stackoverflow.com/questions/64499367/what-are-overlapping-subproblems-in-dynamic-programming-dp (accessed Feb. 19, 2023).

[3] "LeetCode - The World's Leading Online Programming Learning Platform," *LeetCode* https://leetcode.com/problems/longest-palindromic-substring/solutions/151144/bottom-up-dp-two-pointers/ (accessed Feb. 24, 2023).