Competitive Programming and Mathematics Society

Further Dynamic Programming

CPMSoc

Welcome

- Mathematics workshops will run every odd-numbered week (3, 5, 7, ...)
- Programming ones will run every even-numbered week (4, 6, 8, ...)
- Slides will be uploaded on our website (unswcpmsoc.com)
- Advertising for ICPC Prelim Contest:

wisetech global

SOUTH PACIFIC ICPC

Preliminary CONTEST

DATES

Early bird registration closes 31st July*
Late registration closes 31st August
Contest is on 3rd September

REGISTRATION INSTRUCTIONS

Fill out the registration form and follow the prompts at forms.office.com/r/7ti9uqjKEe.

*Early bird:
Free T-shirts for teams that register during early bird only.

Attendance form :D

CPMSOC

Workshop Overview

- DP on a grid
- Subset Sum
- Knapsack
- Questions

Questions

- https://leetcode.com/problems/minimum-path-sum/
- https://leetcode.com/problems/partition-equal-subset-sum/description/
- https://www.hackerrank.com/contests/srin-aadc03/challenges/classic-01-knapsack/problem
- https://leetcode.com/problems/ones-and-zeroes/

Structure of a DP

CPMSOC

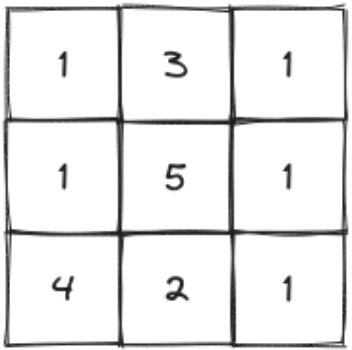
Arc CLUBS SUPPORTED BY Arc INDEPENDENTLY RUN

Steps for Solving DP Problems

1. Subproblems
2. Recurrence
3. Base Cases

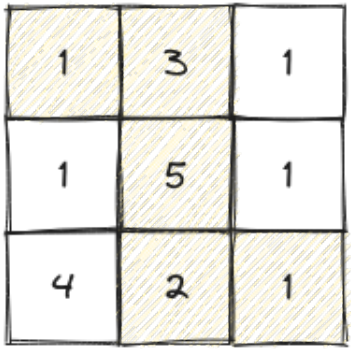
Maximum Grid Path

Given an n x n grid filled with integers, find a path from top left to bottom right, which maximises the sum of all numbers along its path.
Note: You can only move either down or right at any point in time.



Maximum Grid Path

Given an n x n grid filled with integers, find a path from top left to bottom right, which maximises the sum of all numbers along its path.
Note: You can only move either down or right at any point in time.

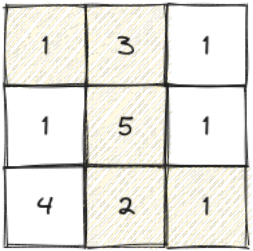


Maximum Grid Path

Given an n x n grid filled with integers, find a path from top left to bottom right, which maximises the sum of all numbers along its path.

Maximum Grid Path

Given an n x n grid filled with integers, find a path from top left to bottom right, which maximises the sum of all numbers along its path.



- What is our current state?

Maximum Grid Path

Given an n x n grid filled with integers, find a path from top left to bottom right, which maximises the sum of all numbers along its path.

- What is our current state?
- What is our base case?

Maximum Grid Path

Given an n x n grid filled with integers, find a path from top left to bottom right, which maximises the sum of all numbers along its path.

- What is our current state?
- What is our base case?
- What is our recurrence?

Subset Sum

CPMSOC

You are given an array of N items, with given weights. Determine if it's possible to select a subset of these items with a total weight of W

Similar question: https://leetcode.com/problems/partition-equal-subset-sum/description/

Subset Sum

CPMSOC

You are given an array of N items, with given weights. Determine if it's possible to select a subset of these items with a total weight of W

Similar question: https://leetcode.com/problems/partition-equal-subset-sum/description/

- What is our current state?

Subset Sum

CPMSOC

You are given an array of N items, with given weights. Determine if it's possible to select a subset of these items with a total weight of W

Similar question: https://leetcode.com/problems/partition-equal-subset-sum/description/

- What is our current state?
- What is our base case?

Subset Sum

CPMSOC

You are given an array of N items, with given weights. Determine if it's possible to select a subset of these items with a total weight of W

Similar question: https://leetcode.com/problems/partition-equal-subset-sum/description/

- What is our current state?
- What is our base case?
- What is our recurrence?

Knapsack

- There are n items, with each having a different volume and value. You only have a bag of capacity m to carry these items. You can only fill up the bag with items until the capacity is full, so what is the most value you can get?

Knapsack

- There are n items, with each having a different volume and value. You only have a bag of capacity m to carry these items. You can only fill up the bag with items until the capacity is full, so what is the most value you can get?
- What is our current state?

Knapsack

- There are n items, with each having a different volume and value. You only have a bag of capacity m to carry these items. You can only fill up the bag with items until the capacity is full, so what is the most value you can get?
- What is our current state?
- What is our base case?

Knapsack

- There are n items, with each having a different volume and value. You only have a bag of capacity m to carry these items. You can only fill up the bag with items until the capacity is full, so what is the most value you can get?
- What is our current state?
- What is our base case?
- What is our recurrence?

Knapsack

- There are n items, with each having a different volume and value. You only have a bag of capacity m to carry these items. You can only fill up the bag with items until the capacity is full, so what is the most value you can get?

Knapsack

- There are n items, with each having a different volume and value. You only have a bag of capacity m to carry these items. You can only fill up the bag with items until the capacity is full, so what is the most value you can get?
- What is our current state?

Knapsack

- There are n items, with each having a different volume and value. You only have a bag of capacity m to carry these items. You can only fill up the bag with items until the capacity is full, so what is the most value you can get?
- What is our current state?
- What is our base case?

Knapsack

- There are n items, with each having a different volume and value. You only have a bag of capacity m to carry these items. You can only fill up the bag with items until the capacity is full, so what is the most value you can get?
- What is our current state?
- What is our base case?
- What is our recurrence?

Knapsack

```
def knapsack(m) {
  if (m == 0) {
    return 0;
  }
  if (memo[m] != -1) {
    return memo[m];
  }
  answer = 0;
  for (item in items) {
    if (volume[item] <= m) {
      answer = max(answer, value[item] + knapsack(m - volume[item]));
    }
  }
  memo[m] = answer;
  return memo[m];
}
```

Improved Knapsack

```
def knapsack(m, n) {
  if (n >= num_items) {
    return 0;
  }
  if (m == 0) {
    return 0;
  }
  if (memo[m][n] != -1) {
    return memo[m][n];
  }
  answer = max(0, knapsack(m, n+1));
  if (volume[n] <= m) {
    answer = max(answer, value[n] + knapsack(m - volume[n], n));
  }
  memo[m] = answer;
  return memo[m];
}
```

Attendance form :D

CPMSOC

Feedback form :D

CPMSOC

Arc CLUBS SUPPORTED BY Arc INDEPENDENTLY RUN

Further events

CPMSOC

Please join us for:

- Maths workshop next week
- Programming workshop in two weeks
- Sign up for ICPC Prelim contest!

wisetech global

SOUTH PACIFIC ICPC

Preliminary CONTEST

DATES

Early bird registration closes 31st July*
Late registration closes 31st August
Contest is on 3rd September

REGISTRATION INSTRUCTIONS

Fill out the registration form and follow the prompts at forms.office.com/r/7ti9uqjKEe.

*Early bird:
Free T-shirts for teams that register during early bird only.