Computer Programming 2 Lab

2023 / 04 / 26

Chen, Yi-Hsuan

Outline

- Dynamic Programming
- Knapsack Problem
- Homework 6
- Exercise 3

Dynamic Programming

Dynamic Programming

SOP

- **建立狀態，找出轉移關係**
  - 有時候時間複雜度太高
- 優化狀態定義
- 優化轉移方式
  - 矩陣快速冪
  - 資料結構優化
  - 分治法優化
  - 凸包優化
  - 四邊形優化

Dynamic Programming

建立狀態，找出轉移關係

1. 列狀態
   $dp(i)$: 以a_i為結尾的max
2. 找出轉移式
   $dp(i) : max_{0 \leq j < i} : dp(j) + a_j$
3. 找答案
   如果時間複雜度太高，優化

Dynamic Programming

Example 1 - LeetCode 70. Climbing Stairs

You are climbing a staircase. It takes n steps to reach the top.
Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

Dynamic Programming

Example 1 - LeetCode 70. Climbing Stairs

$O(n)$

1. $dp(i)$: The ways to climb i steps.
2. $dp(i) = dp(i-1) + dp(i-2)$
3. $ans = dp(n)$

Knapsack Problem

Knapsack Problem

將一些東西放入背包中，使背包裡的物品總價值最高。

- Fractional Knapsack Problem
- 0/1 Knapsack Problem
- Unbounded Knapsack Problem
- Bounded Knapsack Problem

Knapsack Problem

Fractional Knapsack Problem

找出價值/重量最高的優先拿。(greedy)

Knapsack Problem

Example 2 - ZeroJudge f627. 1. 礦砂採集

Knapsack Problem

0/1 Knapsack Problem

每個物品要或不要。

- ~~greedy~~ (假解)
  5 4 4(weight)
  9 4 5(value)

Knapsack Problem

0/1 Knapsack Problem

$O(2^n \cdot n)$

- 窮舉所有子集合
  - 所有子集合$O(2^n)$個
  - 驗證一個子集合$O(n)$

Knapsack Problem

0/1 Knapsack Problem

time $O(n \cdot w)$, space $O(w)$

1. $dp(i)$: exactly i weight in bags max value
2. $dp(i)$: $max\{dp(i - w_k) + v_k\}$ $(1 \le k \le n, w_k \le i)$
    - 必須由大到小遍歷
3. $ans = max\{dp(i)\}$

Knapsack Problem

Example 3 - ZeroJudge a587. 祖靈好孝順 ˋˇˊ

Knapsack Problem

Unbounded Knapsack Problem

一種物品可以一直拿。

time $O(n \cdot w)$, space $O(w)$

1. $dp(i)$: exactly i weight in bags max value
2. $dp(i)$: $max\{dp(i - w_k) + v_k\}$
3. $ans = max\{dp(i)\}$

Knapsack Problem

Example 4 - ZeroJudge e574. 10404 - Bachet's Game

Knapsack Problem

Bounded Knapsack Problem

每種物品有數個 -> 0/1 Knapsack Problem

time $O(n \cdot w)$, space $O(w)$

1. $dp(i)$: exactly i weight in bags max value
2. $dp(i) : max\{dp(i - w_k) + v_k\}\ (1 \leq k \leq n, w_k \leq i)$
3. $ans = max\{dp(i)\}$

Homework 6 - Subsequence Addition

Homework 6

Description

Initially, array a contains just the number 1. You can perform several operations in order to change the array. In an operation, you can select some subsequence of a and add into a an element equal to the sum of all elements of the subsequence.

You are given a final array c. Check if c can be obtained from the initial array a by performing some number (possibly 0) of operations on the initial array.

A sequence b is a subsequence of a sequence a if b can be obtained from a by the deletion of several (possibly zero, but not all) elements. In other words, select k $(1 \leq k \leq |a|)$ distinct indices $i_1, i_2, \dots, i_k$ and insert anywhere into a a new element with the value equal to $a_{i_1} + a_{i_2} + \cdots + a_{i_k}$.

Homework 6

Input

The first line of the input contains an integer t $(1 \le t \le 1000)$ — the number of test cases. The description of the test cases follows.

The first line of each test case contains a single integer n $(1 \le n \le 2 \cdot 10^5)$ — the number of elements the final array c should have.

The second line of each test case contains n space-separated integers c_i $(1 \le c_i \le 2 \cdot 10^5)$ — the elements of the final array c that should be obtained from the initial array a.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Homework 6

Output

For each test case, output "YES" (without quotes) if such a sequence of operations exists, and "NO" (without quotes) otherwise.

Homework 6

Sample1

Input sample	Output sample
6 1 1 1 2 5 5 1 3 2 1	YES NO YES NO YES YES

Input sample	Output sample
5 7 1 5 2 1 3 1 1 1 5 1 1 4 2 1	

Homework 6

Constraints

For 30%:

- $1 \leq n, c_i \leq 100$

For 60%:

- $1 \leq n, c_i \leq 5000$

For 100%:

- $1 \leq t \leq 1000$
- $1 \leq n, c_i \leq 2 \cdot 10^5$

Exercise 3 - Snake Line

Exercise 3

Description

There are n snakes to be arranged in a straight line of length m, and the length of each snake is a_i, is it possible?

Exercise 3

Input

Input consists of multiple test cases.
The first line contains a single integer t, the number of test cases $(1 \leq t \leq 10)$.
Each test case has three lines, the first line contains an integer m $(0 \leq m \leq 10^5)$.
The second line has an integer n $(1 \leq n \leq 1000)$.
The third line has n integers. The i-th integer represents the length a_i of the i-th snake.

Exercise 3

Output

For each test case, output YES if possible, output NO if not possible.

Exercise 3

Sample1

<table>
<tr><th>Input sample</th><th>Output sample</th></tr>
<tr><td>3
25
4
10 12 5 7
925
10
45 15 120 500 235 58 6 12 175 70</td><td>NO
YES
NO</td></tr>
</table>

Input sample	Output sample
120 5 25 25 25 25 25	

Exercise 3

Constraints

For 30%:

- $1 \le a_i \le m \le 10^3$
- $1 \le n \le 20$

For 60%:

- $1 \le a_i \le m \le 10^4$
- $1 \le n \le 100$

For 100%:

- $1 \le t \le 10$
- $1 \le a_i \le m \le 10^5$
- $1 \le n \le 1000$