

Sweet Tapas

4. Bucky Badger is having dinner at an upscale tapas bar, where he will order many small plates. There are N plates of food on the menu, where information for plate i is given by a triple of non-negative integers (v_i, c_i, s_i) : the plate's volume v_i , calories c_i , and sweetness $s_i \in \{0, 1\}$ (the plate is sweet if $s_i = 1$ and not sweet if $s_i = 0$). Bucky is on a **diet**: he wants to eat no more than C calories during his meal, but wants to fill his stomach as much as possible. He also wants to order exactly $S < N$ sweet plates, without purchasing the same dish twice.

Describe an $O(NCS)$ -time algorithm to find the maximum volume of food Bucky can eat given his diet.

Solution:

BC:

$dp[0][c][0] = 0$
when 0 plates left

$dp[0][c][s] = -\infty$
not exactly s plates then

↑ 0 20 40 | 80
40 —
20 40 20 —
70 60 50 40 30 20 10 —

$$dp[i][c][s] = \max \left(v_i + dp[i-1][c-c_i][s-s_i], dp[i-1][c][s] \right)$$

↑ $i=0 \dots N$ take don't take

none to next plate this many cal's left now {0,1}

total maximum
volume of food

Calories $\leq C$ and
order only $S < N$ sweet
plates

Name: _____

Wisc id: _____

Coin Change (again)

1. *CLRS 3rd edition (p. 446)*. Consider the problem of making change for n cents using the fewest number of coins. Assume that each coin's value is an integer. Give an $O(nk)$ -time algorithm that makes change for any set of k different coin denomination, assuming one of the coins is a penny.

answer represent
fewest amount
of coins to
total n

Solution:

 $k = 4$ $n = 30$

1¢ 5¢ 10¢ 25¢

 $k = 3$

10¢ 20¢ 30¢ $n = 40$

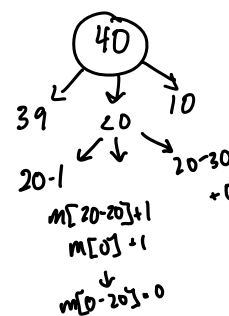
for j from 1 to 30

$$m[40] = \min(m[i - v_j]) + 1$$

$$m[40] = \min(m[40 - 1]) + 1 \rightarrow \min[39] + 1$$

$$m[40] = \min(m[40 - 20]) + 1 \rightarrow \min[20] + 1$$

$$m[40] = \min(m[40 - 30]) + 1 \rightarrow \min[10] + 1$$



$$v_j = [1, 20, 30]$$

$$m[40] = \min(m[40 - v_j]) + 1;$$

$$\min(m[40], m[40 - v_j]) + 1$$

min # of
coins to achieve
 i

$$m[i] = \min(m[i], m[i - v_j]) + 1$$

$j = 1 \dots k$

k different
coin denominations

BC: $i - v_j < 0$
 $m[i] = 0$

$O(nk)$
 \uparrow
 n grids $\leftarrow k$ coins

the answer will be located at $m[n]$ where n is the goal cents

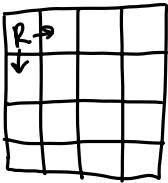
Coin Collecting

2. Several coins of different value are placed on a $m \times n$ board. Let $c_{i,j} \geq 0$ be the value of the coin placed on grid- (i, j) . Note that $c_{i,j} = 0$ implies that there is no coin on grid- (i, j) .

You placed a robot on the board to collect the coins. The robot starts at the top-left corner of the board (grid- $(0, 0)$), but it can only move right or down. What is the maximum total value of coins the robot can collect?

left or up

Solution:
n



m

goal grid- (m, n) b/c can't move right or down anymore.
find max from each path of choosing either right or down
C each step.
basically this off of the previous choice that it's made
so start at $g(m, n)$ go from there

left

up

$$m[i, j] = \max(m[i, j-1] + c_{i,j}, m[i-1, j] + c_{i,j})$$

vs

$$m[i, j] = \max(m[i, j-1], m[i-1, j]) + c_{i,j}$$

$$\text{answer} = m[m, n]$$

bc: edge / corner
 $m[i, j] = m[i, j]$

m: 1...i

n: 1...j

Knapsack

3. (0-1 Knapsack) Given n items where i -th item has value v_i and weighs w_i , output the maximum value for a knapsack with capacity of W . Note that each item can be chosen at most once.

Solution:

$$dp[i][w] = \max(dp[i-1][w], v_i + dp[i-1][w-w_i])$$

not pick i

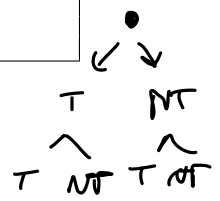
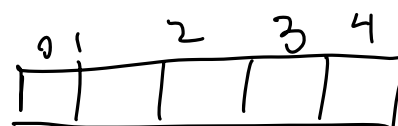
take i

max total

val considering 1...i

w weight $\leq W$

iterate through all items
& choose if pick v_i or not
answer @



collect items
to be as close
to W without
exceeding

bc:
(1) $i \leq 0$ (no items left)
 $m[i, j] = 0$
value = 0
for non-existing
item
 $m[i, j] = 0$
(2) $w \leq 0$ (capacity = 0)
 $w - w_i \leq 0$
 $m[i, j] = 0$

4. (Unbounded Knapsack) Given n items where i -th item has value v_i and weighs w_i , output the maximum value for a knapsack with capacity of W . Note that each item can be chosen **unlimited times**.

Solution:

n items: $v_1, v_2, v_3, \dots, v_n$
 $w_1, w_2, w_3, \dots, w_n$ $m[v, w]$

$dp[w] = \max_{j=1 \dots n} (dp[w], v_j + dp[w - w_j])$

↑
 sum capacity

max total value w/ weight $\leq w$
 answer $dp[w]$

40
 59 20 10

create a 1D array where last item holds the answer

w_1, w_2, w_3, w_n, W

↑
 makes it like coin problem

5. (Multidimensional Knapsack) Given n items where i -th item has value v_i , weighs w_i , and has size d_i , output the maximum value for a knapsack with capacity of W and size of D . Note that each item can be chosen at most once.

Solution:

$dp[i][w][d] = \max \left(\overset{\text{don't take}}{dp[i-1][w][d]}, \overset{\text{take}}{v_i + dp[i-1][w - w_i][d - d_i]} \right)$

↑
 weight $\leq w$
 size $\leq d$

$dp[0][w][d] = 0$

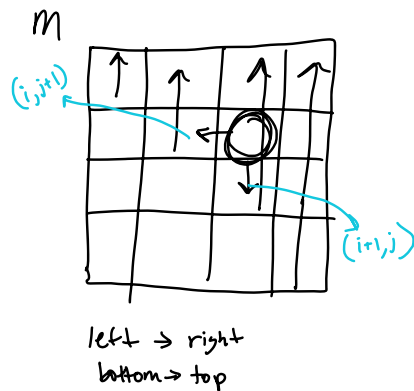
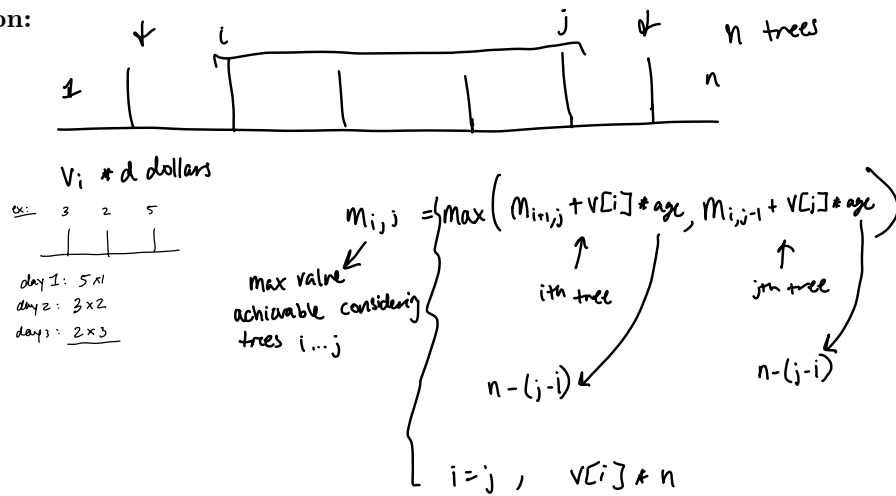
Farmer

6. A farmer has a row of n trees that grow fruits, and he wants to earn some money by cutting down the trees and selling the fruits. He is old so he only cuts down one tree per day. Also, each day he only cuts down the leftmost tree or the rightmost tree.

Initially, cutting down the i -th tree allows the farmer to earn $v_i > 0$ dollars. As time goes by, a fruit might rot, so cutting it down earlier allows the farmer to earn more! Cutting down the i -th tree on the d -th day allows the farmer to earn $v_i \times (n - d + 1)$ dollars.

What is the maximum amount of money the farmer could achieve?

Solution:



$$O(n^2) + O(1)$$

counter:
5 1 1 1 9 9 4

4x1	5x1
5x2	1x2
1x3	1x3
1x4	1x4
1x5	4x5
26	34 ✓

\therefore choosy min won't work

Name: _____

Wisc id: _____

Robber

1. You are a robber that plans to rob n houses along a street. Each house stashes some amount of money $v_i > 0$. However, you cannot rob two adjacent houses since this will alert the police. What is the maximum amount of money you can rob without alerting the police?

Solution:

$$M[i] = \max \left(\underbrace{v_i + M[i-2]}_{\substack{\text{take} \\ \downarrow}}, \underbrace{M[i-1]}_{\substack{\text{don't take} \\ \downarrow}} \right)$$

$$i = 1 \dots n$$

$$M = \underbrace{v_1 \quad v_2 \quad v_3 \quad v_4 \quad v_5}_{v_1 + v_3 + v_5}$$

$$M[i-2] + v_i$$

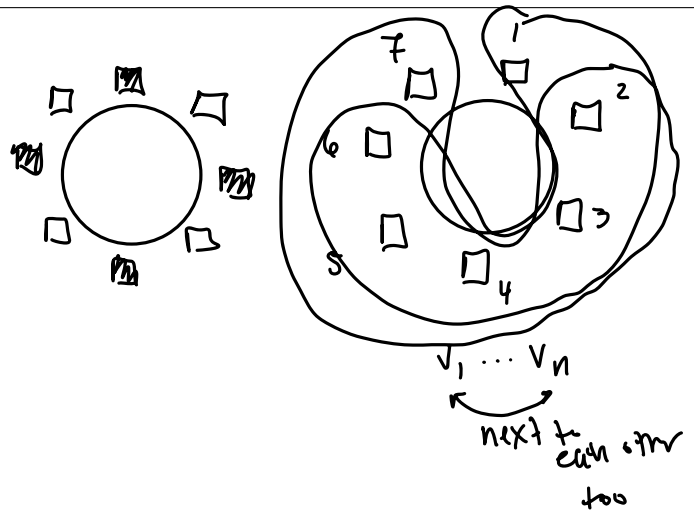
$$\uparrow$$

 max amt
 of money
 can rob

2. What if the houses are arranged in a circle?

Solution:

max between take 1 - n-1
 and
 2 - n



Convex Polygon Triangulation

3. You have a convex n -sided polygon where each vertex has an integer value.

You will triangulate the polygon into $n - 2$ triangles. For each triangle, the value of that triangle is the product of the values of its vertices, and the total score of the triangulation is the sum of these values over all $n - 2$ triangles in the triangulation.

Return the smallest possible total score that you can achieve with some triangulation of the polygon.

Solution:

Sweet Tapas

4. Bucky Badger is having dinner at an upscale tapas bar, where he will order many small plates. There are N plates of food on the menu, where information for plate i is given by a triple of non-negative integers (v_i, c_i, s_i) : the plate's volume v_i , calories c_i , and sweetness $s_i \in \{0, 1\}$ (the plate is sweet if $s_i = 1$ and not sweet if $s_i = 0$). Bucky is on a **diet**: he wants to eat no more than C calories during his meal, but wants to fill his stomach as much as possible. He also wants to order exactly $S < N$ sweet plates, without purchasing the same dish twice.

Describe an $O(NCS)$ -time algorithm to find the maximum volume of food Bucky can eat given his diet.

Solution:

$$(v_1, c_1, s_1) \leq C$$

$$(v_2, c_2, s_2) \quad \text{S can't be 1 for all } N \text{ plates}$$

max vol 1..i, calories $\leq C$, exactly s sweet plates.

$$dp[i][c][s] = \max \left(dp[i-1][c][s], v_i + dp[i-1][c-c_i][s-s_i] \right)$$

$$\text{answer} = dp[N][C][S]$$

don't take item

take item

$$-s_i \leq 0$$

$$dp[v_i][c_i][s_i]$$

$$S < N$$

$$s < N$$

$$dp[0][c][0] = 0$$

$$dp[0][c][s] = -\infty$$

not exactly s sweet plates

if

$$s = \{0, 1, 1, 0\} = 2$$

$$v_0$$

LIS

5. Given an integer array, return the length of the longest strictly increasing subsequence. Can you come up with a $O(n \log n)$ solution?

Solution: