

# Discrete Math — Homework 6 Solutions

Yuquan Sun, SID 10234900421

April 8, 2025

## Q1

Construct a sequence of 16 positive integers that has no increasing or decreasing subsequence of five terms.

**Solution:** 4,3,2,1,8,7,6,5,12,11,10,9,16,15,14,13.

## Q2

What is the coefficients of  $x^8, x^9$  and  $x^{10}$  in  $(2 - x)^{19}$ ?

**Solution:**

$$\begin{aligned}(2 - x)^{19} &= \cdots + \binom{19}{8} \cdot 2^{11} \cdot (-1)^8 \cdot x^8 \\ &\quad + \binom{19}{9} \cdot 2^{10} \cdot (-1)^9 \cdot x^9 \\ &\quad + \binom{19}{10} \cdot 2^9 \cdot (-1)^{10} \cdot x^{10} \\ &\quad + \cdots\end{aligned}$$

## Q3

Let  $n$  be a positive integer. Show that

$$\binom{2n}{n+1} + \binom{2n}{n} = \binom{2n+2}{n+1} / 2$$

**Proof:** From the property of combination:  $\binom{n}{k} + \binom{n}{k-1} = \binom{n+1}{k}$ , we deduce:

$$\begin{aligned}\binom{2n}{n+1} + \binom{2n}{n} &= \binom{2n+1}{n+1} \\ &= \frac{1}{2} \left[ \binom{2n+1}{n} + \binom{2n+1}{n+1} \right] \\ &= \frac{1}{2} \binom{2n+2}{n+1}\end{aligned}$$

□

## Q4

Give a combinatorial proof that

$$\sum_{k=1}^n k \binom{n}{k} = n \cdot 2^{n-1}$$

**Proof:** Let's count in two ways the number of ways to select a committee and to then select a leader of the committee out of  $n$  people.

First, if we consider the committee by the scale of people. For a  $k$ -people committee, then the cases are first choose  $k$  people, namely  $\binom{n}{k}$ ; then there are  $k$  chances to pick a leader out of all of them, and thus, the answer should be  $k \binom{n}{k}$ . Let  $k$  iterate from 1 to  $n$ , then the total sum should be  $\sum_{k=1}^n k \binom{n}{k}$ .

From a different perspective, we first pick a leader then there are  $n$  choices in total, then for the rest  $n - 1$  people, they are either in the committee or not, so there are  $2^{n-1}$  cases, put the 2 parts together, we get  $n \cdot 2^{n-1}$ .  $\square$

## Q5

In how many ways can a  $2 \times n$  rectangular checkerboard be tiled using  $1 \times 2$  and  $2 \times 2$  pieces?

**Solution:** Denote  $F[n]$  as the answer for the  $2 \times n$  case.

- (I) Base case  $n = 1$ .  $F[1] = 1$ .
- (II) Base case  $n = 2$ .  $F[2] = 3$ .
- (III) Common case for  $n \geq 3$ .  
 If we use  $1 \times 2$  blocks, then there are 2 ways. If put it **vertically**, then  $F[n - 2]$ ; if put it **horizontally**, then  $F[n - 1]$ .  
 If we use  $2 \times 2$  block, then  $F[n - 2]$ .  
 In total,  $F[n] = F[n - 1] + 2F[n - 2]$ .

Solving it, we get  $F[n] = \frac{2}{3} \cdot 2^n + \frac{1}{3} \cdot (-1)^n$ .

## Q6

- (a) Find the recurrence relation satisfied by  $R_n$ , where  $R_n$  is # regions that a plane is divided into by  $n$  lines, if no two of the lines are parallel and no three of the lines go through the same point.

**Solution:** It's easy to see that  $R_n = R_{n-1} + n$  where  $R_1 = 2$ .

- (b) Find  $R_n$  using iteration.

**Solution:**  $R_n = R_{n-1} + n = R_{n-2} + (n-1) + n = R_1 + 2 + \cdots + (n-1) + n = 1 + n(n+1)/2$ .

## Q7

A vending machine dispensing books of stamps accepts only one-dollar coins, \$1 bills, and \$5 bills.

- (a) Find a recurrence relation for the number of ways to deposit  $n$  dollars in the vending machine, where the order in which the coins and bills are deposited matters.

**Solution:** Denote  $F[n]$  as the ways for the  $n$  dollars case. Considering always put the newest item at the end of the sequence, then this ensures iterate all permutations but no repetition.

$$\text{Then } \begin{cases} F[n] = 2F[n-1] & , 0 < n < 5 \\ F[n] = 2F[n-1] + F[n-5] & , n \geq 5 \end{cases}.$$

- (b) What are the initial conditions?

**Solution:**  $F[0] = 1$ .

- (c) How many ways are there to deposit \$10 for a book of stamps?

**Solution:** By calculation, we get **Ans** = 1217.

## Q8

For bit strings, find a recurrence relation for the number of bit strings of length  $n$  that contain an odd number of 0s.

**Solution:** Denote  $F[n, 0]$  as the number of cases where there exists even 0s, and  $F[n, 1]$  for the odd case. Then we can get the recurrence relation:

$$F[n, 0] = F[n-1, 0] + F[n-1, 1]$$

$$F[n, 1] = F[n-1, 0] + F[n-1, 1]$$

And the initial condition,  $F[1, 0] = F[1, 1] = 1$ .

## Q9

Given two strings  $A$  and  $B$ , we need to find the minimum number of operations which can be applied on  $A$  to convert it to  $B$ . The operations are:

- Edit - Change a character to another character;
- Delete - Delete a character;
- Insert - Insert a character.

The **edit distance** of two strings is defined by the minimum # operations required to transform one string into the other. For the following two strings, their edit distance is 3.

*GC***G***TATG***A***GGCTA*–*ACGC*  
*GC*–*TATG***C***GGCTA***T***ACG*

Please utilize the dynamic programming to compute the edit distance between two strings  $A$  and  $B$ .

- (a) Define the subproblems for DP;  
 Denote  $F[i, j]$  as the minimum # operations required to let the former  $i$  letters of string  $A$  and former  $j$  letters of string  $B$  be the same.

(b) Find the recurrence;

Denote  $A[i]$  as the  $i$ th letter of string  $A$ .

If  $A[i] = B[j]$ , then there is no need to make any change and thus  $F[i, j] = F[i - 1, j - 1]$ . If  $A[i] \neq B[j]$ , we have 3 approaches:

- **Edit** Then we just need to ensure the former letters are the same aka.  $F[i, j] = \min \{F[i, j], F[i - 1, j - 1] + 1\}$ .
- **Delete** Then we should make sure the former letters of  $A$  should be same as  $B[1 \cdots j]$  aka.  $F[i, j] = \min \{F[i, j], F[i - 1, j] + 1\}$ .
- **Insert** Then we should make sure the letters of  $A$  is the same as the former letters of string  $B$  aka.  $F[i, j] = \min \{F[i, j], F[i, j - 1] + 1\}$ .

To sum up, we have:

$$F[i, j] = \begin{cases} F[i - 1, j - 1] & , A[i] = B[j] \\ \min \{F[i - 1, j - 1], F[i - 1, j], F[i, j - 1]\} + 1 & , A[i] \neq B[j] \end{cases}$$

Initial condition:

$\forall i \in \{1, 2, \dots, A.length\}, F[i, 0] = i$  and  $\forall i \in \{1, 2, \dots, B.length\}, F[0, i] = i$ .

(c) Implement the algorithm to return the edit distance of two strings.

---

**Algorithm 1** edit distance of string  $A$  and string  $B$

---

**Require:** string  $A, B$ , length of string  $A, B$   $l_A, l_B$

**Ensure:** the edit distance between string  $A$  and  $B$   $F[l_A, l_B]$ .

```

for  $i \leftarrow 1$  to  $l_A$  do
   $F[i, 0] = i$ 
for  $i \leftarrow 1$  to  $l_B$  do
   $F[0, i] = i$ 
for  $i \leftarrow 1$  to  $l_A$  do
  for  $j \leftarrow 1$  to  $l_B$  do
    if  $A[i] = B[j]$  then
       $F[i, j] = F[i - 1, j - 1]$ 
    else
       $F[i, j] = \min (F[i - 1, j - 1], F[i - 1, j], F[i, j - 1]) + 1$ 
   $\triangleright$  return  $F[l_A, l_B]$ 

```

---

## Q10

In the knapsack problem we are given a set of  $n$  items, where each item  $i$  is specified by a size  $s_i$  and a value  $v_i$ . We are also given a size bound  $S$  (the size of our knapsack). The goal is to find the subset of items of maximum total value such that sum of their sizes is at most  $S$  (they all fit into the knapsack). To implement a DP algorithm to solve this problem,

(a) Define subproblems;

Denote  $F[i, j]$  as the maximum value under the condition where we just use the former  $i$  items and the sum of the size doesn't exceed  $j$ .

- (b) Find the recurrence relation;

For the  $i$ th item, we can either choose or not. Thus, we get

$$F[i, j] = \begin{cases} F[i-1, j] & , \text{ if } s_i > j \\ \max \{F[i-1, j], F[i-1, j-s_i] + v_i\} & , \text{ if } s_i \leq j \end{cases}$$

- (c) Solve the base cases;

$$\forall i \in \{0, 1, 2, \dots, n\}, F[i, 0] = 0.$$

- (d) Implement the algorithm to return the solution of the knapsack problem.

---

**Algorithm 2** 0-1 knapsack problem
 

---

**Require:** number of item  $n$ , max capacity  $m$ , item sizes  $s[1 \cdots n]$ , item value  $v[1 \cdots n]$ .

**Ensure:** max value with sizes not exceed capacity  $F[n, m]$ .

```

for  $i \leftarrow 0$  to  $n$  do
   $F[i, 0] \leftarrow 0$ 
for  $i \leftarrow 1$  to  $n$  do
  for  $j \leftarrow 0$  to  $m$  do
    if  $s[i] > j$  then
       $F[i, j] = F[i-1, j]$ 
    else
       $F[i, j] = \max(F[i-1, j], F[i-1, j-s[i]] + v[i])$ 
   $\triangleright$  return  $F[n, m]$ 

```

---

**Q11**

Solve the following recurrence relations

- (a)
- $a_n = 5a_{n-1} - 6a_{n-2}$
- for
- $n \geq 2$
- with
- $a_0 = 1$
- and
- $a_1 = 0$
- .

**Solution:** This is a LHR and its

- (b)
- $a_n = a_{n-1} + 6a_{n-2}$
- for
- $n \geq 2$
- with
- $a_0 = 6$
- and
- $a_1 = 8$
- .

**Q12**

The Lucas numbers satisfy the recurrence relation

$$L_n = L_{n-1} + L_{n-2}$$

and the initial conditions  $L_0 = 2$  and  $L_1 = 1$ .

- (a) Show that  $L_n = f_{n-1} + f_{n+1}$  for  $n = 2, 3, \dots$ , where  $f_n$  is the  $n$ -th Fibonacci number.
- (b) Find an explicit formula for the Lucas numbers.