Discrete Math — Homework 6 Solutions

Yuquan Sun, SID 10234900421

April 12, 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:

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

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:

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

$\square$

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 1×2 and 2×2 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 1×2 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 2×2 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.
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 $\textbf{Ans} = 1217$.

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:

a. Edit - Change a character to another character;

b. Delete - Delete a character;

c. 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.

$$GCGTATGAGGCTA-ACGC$$
$$GC-TATGCGGCTATACG$$

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

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

```
for i ← 1 to l_A do
    F[i, 0] = i
for i ← 1 to l_B do
    F[0, i] = i
for i ← 1 to l_A do
    for j ← 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        ▷ 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 ith 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;
$\forall i \in \{0,1,2,\ldots n\}, F[i,0] = 0$.

(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 ← 0 to n do
    F[i,0] ← 0
for i ← 1 to n do
    for j ← 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])        ▷ 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: The characteristic equation of $a_n = 5a_{n-1} - 6a_{n-2}$ is $\lambda^2 - 5\lambda + 6 = 0$.

Solving it, we get: $\lambda_1 = 2, \lambda_2 = 3$.

So the general solution of the recurrence relation should be $a_n = c_1 \cdot 2^n + c_2 \cdot 3^n$.

The initial condition means $\begin{cases} c_1 + \ \ c_2 = 1 \\ 2c_1 + 3c_2 = 0 \end{cases}$, then we get $\begin{cases} c_1 = \ \ \ 3 \\ c_2 = -2 \end{cases}$.

Therefore, $a_n = 3 \cdot 2^n + 2 \cdot 3^n$.

(b) $a_n = a_{n-1} + 6a_{n-2}$ for $n \geq 2$ with $a_0 = 6$ and $a_1 = 8$.
Solution: The characteristic equation of $a_n = a_{n-1} + 6a_{n-2}$ is $\lambda^2 - \lambda - 6 = 0$.

Solving it, we get: $\lambda_1 = 3, \lambda_2 = -2$.

So the general solution of the recurrence relation should be $a_n = c_1 \cdot (-2)^n + c_2 \cdot 3^n$.

The initial condition means $\begin{cases} c_1 + \ \ c_2 = 6 \\ -2c_1 + 3c_2 = 8 \end{cases}$, then we get $\begin{cases} c_1 = 2 \\ c_2 = 4 \end{cases}$.

Therefore, $a_n = 2 \cdot (-2)^n + 4 \cdot 3^n$.

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 $n = 2, 3, \cdots$, where f_n is the $n-$th Fibonacci number.
Proof: Consider using strong inductive method.

(I) **Base case** $n = 2$, then $L_2 = L_0 + L_1 = 3 = f_1 + f_3$.

(II) **Inductive step** If it holds for $n = 2, 3, \dots, k$, aka. $\forall i \in \{2, 3, \dots, k\}, L_i = f_{i-1} + f_{i+1}$. Then for the $n = k+1$ case, $L_{k+1} = L_k + L_{k-1} = f_{k-1} + f_{k+1} + f_{k-2} + f_k = f_k + f_{k+2}$.

Then, use the strong inductio method, we conclude $L_n = f_{n-1} + f_{n+1}$. $\square$

(b) Find an explicit formula for the Lucas numbers.
Solution: Since we know the general solution for the Fibonacci number is

$$f_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$$

Then, use the conclution from (a) we get

$$\begin{aligned} L_n &= f_{n-1} + f_{n+1} \\ &= \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^{n-1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n-1} + \left(\frac{1+\sqrt{5}}{2}\right)^{n+1} - \left(\frac{1-\sqrt{5}}{2}\right)^{n+1}\right] \\ &= \frac{\sqrt{5}+1}{2} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^{n-1} - \frac{\sqrt{5}-1}{2} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^{n-1} \\ &= \left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{1-\sqrt{5}}{2}\right)^n \end{aligned}$$