CSE 21 – Fall 2019 – Practice Final Solution

1. **Order questions.** For each of the following pairs of functions f and g, choose whether $f \in o(g), f \in \Theta(g)$, or $f \in \omega(g)$ (exactly one will be true).

   (a) $f(n) = n, g(n) = 2^{\lfloor \log n \rfloor}$

   (b) $f(n) = \log(n^4), g(n) = (\log n)^4$.

   (c) $f(n) = n^2, g(n) = 1000n^2 + 2^{100}$

   (d) $f(n) = 2^{2n}, g(n) = 2^n$.

   (e) $f(n) = n!, g(n) = n^n$.

2. **Analyzing algorithms.** What is the order of the standard multiplication algorithm for multiplying two $n \times n$ matrices? Suppose that any two matrix entries can be multiplied together in constant time, and that addition takes constant time.

3. **Recursive algorithms.** Here's a recursive algorithm, that given an n bit number in binary, $x = x_{n-1} \dots x_0$ and an $m < n$ bit number in binary y, computes $x \mod y$. It uses the following sub-routines: $Add(x, y)$ adds two n bit binary numbers in $O(n)$ time, $AsLarge(x, y)$ returns true if and only if $x \geq y$ and also takes $O(n)$ time, and $Subtract(x, y)$ computes $x - y$ and is also $O(n)$ time. We assume $y \geq 2$ and $n \geq 1$.

   Mod $(x = x_{n-1} \dots x_1 x_0,\ y)$

   1. IF $n = 1$, return x_0.
   2. $v \leftarrow Mod(x_{n-1} \dots x_1,\ y)$.
   3. $v \leftarrow Add(v, v)$
   4. $v \leftarrow Add(v, x_0)$.
   5. IF $AsLarge(v, y)$ THEN $v \leftarrow Subtract(v, y)$.
   6. Return v

   (a) **Correctness.** Prove by induction on n that $Mod(x, y)$ returns $x \mod y$.
   Hints:
   - Let $x = 2a + x_0$, where $a = x_{n-1} \dots x_1$.
   - To show that two numbers are the same mod y, show that their difference is a multiple of y.

   (b) **Recurrence.** Give a recurrence for $T(n)$, the time this algorithm takes on an input of size n.

   (c) **Solving recurrences.** Solve this recurrence to give the Big O time of this algorithm.

5. **Recursive Counting**

For problems (a)-(e), it is not required to solve the recurrence:

   (a) In a round-robin tennis tournament with n players, every tennis player plays against every other player. Let $T(n)$ be the total number of tennis matches taking place among the n players. Find a recurrence for $T(n)$ and explain in words why $T(n)$ satisfies this recurrence.

(b) In a board game, players must place colored tiles in a single row. There are red 1×1 tiles, yellow 2×1 tiles, blue 2×1 tiles, purple 3×1 tiles, and green 3×1 tiles. Let $C(n)$ be the number of different $n \times 1$ colored rows that can be created using these tiles. Find a recurrence for $C(n)$ and explain in words why $C(n)$ satisfies this recurrence.

(c) Any binary string can be broken into contiguous chunks of the same character, called runs. For example, 001110100000 has:

- a run of 0s of length 2
- a run of 1s of length 3
- a run of 0s of length 1
- a run of 1s of length 1
- a run of 0s of length 5

0 $\boxed{B(n-1)}$

1 $\boxed{}$

Let $B(n)$ be the number of length n binary strings where each run of 1s has even length. Find a recurrence for $B(n)$ and explain in words why $B(n)$ satisfies this recurrence.

(d) Let $S(n)$ be the number of subsets of $\{1, 2, \ldots, n\}$ having the following property: there are no three elements in the subset that are consecutive integers. Find a recurrence for $S(n)$ and explain in words why $S(n)$ satisfies this recurrence.

4. Counting. (a) How many 5-card hands can be formed from an ordinary deck of 52 cards if exactly two suits are present in the hand?

(b) In any bit string, the longest consecutive run length is the maximum number of consecutive 1's or consecutive 0's in the string. For example, in the string 1101000111, the longest consecutive run length is 3. How many bit strings of length 10 have a longest consecutive run length of 6?

(c) A software company assigns its summer interns to one of three divisions: design, implementation, and testing. In how many ways can a group of ten interns be assigned to these divisions if each division needs at least one intern?

(d) How many numbers in the interval [1,10000] are divisible by 7, 9, or 11?

(e) A California license plate follows the format $DLLLDDD$, where D represents a digit and L represents an upper case letter. For example, a valid California license plate is $5JBC434$ (Quang's plate). How many California license plates are possible?

Under a fixed-length encoding scheme, how many bits of memory does the California DMV need to allocate in its database to store each license plate number?

5. Probability. (a) I have 10 shirts, 6 pairs of pants, and 3 jackets. Every day I dress at random, picking one of each category. What is the probability that today I am wearing at least one garment I was wearing yesterday?

(b) A permutation of size n is a rearrangement of the numbers $\{1, 2, \ldots, n\}$ in any order. A *rise* in a permutation occurs when a larger number immediately follows a smaller one. For example, if $n = 5$, the permutation 1 3 2 4 5 has three rises. What is the expected number of rises in a permutation of size n?

(c) There are n teams in a sports league. Over the course of a season, each team plays every other team exactly once. If the outcome of each game is a fair random coin flip, and there are no ties, what is the probability that some team wins all of its games?

(d) Suppose there are 10 people of distinct heights standing in line for free food, arranged in a random order. Use linearity of expectation to calculate the expected number of people who can see all the way to the front of the line.

(e) Assume that every time you go on a job interview, your chance of getting a job offer is 25%. How many job interviews must you go on so that the probability of your getting a job offer is greater than 95%?

(f) Suppose that you roll two dice and don't get to look at the outcome. Your friend looks at the outcome and tells you honestly that at least one of the dice came up 6. What is the probability that the sum of your two dice is 8?

(g) Suppose there are n people assigned to m different tasks. Assume that each person is randomly assigned a task and that for each person, all tasks are equally likely. Use linearity of expectation to find the expected number of people working on a task alone.

(h) Tesla is planning to introduce a new self-driving car. The company commissions a marketing report for each new car that predicts either the success or the failure of the car. Of the new cars introduced by the company so far, 65% have been successes. Furthermore, 70% of their successful cars were predicted to be successes, while 40% of failed cars were predicted to be successes. Find the probability that this new self-driving car will be successful if its success has been predicted.

(i) Say we have a coin that has probability p of landing on Heads, and probability $1 - p$ of landing on Tails. Let X_n be the random variable that indicates the number of heads in n independent tosses of this coin. What is the variance of X_1? What is the variance of X_n?

6. Representing problems as graphs. You have a system of n variables representing real numbers, $X_1, \ldots, X_n$. You are given a list of inequalities of the form $X_i < X_j$ for some pairs i and j. You want to know whether you can deduce with certainty from the given information that $X_1 < X_n$.

For example, say n is 4. A possible input is the list of inequalities $X_1 < X_2, X_1 < X_3, X_4 < X_3$ and $X_3 < X_2$. Does it follow that $X_1 < X_4$?

(a) Give a description of a directed graph that would help solve this problem. Be sure to define both the vertices and edges in terms of the variables and known inequalities.

(b) Draw the graph you described for the example above. Does $X_1 < X_4$ follow? Why or why not?

(c) Say which algorithm from lecture we could use on such a graph to determine whether $X_1 < X_n$ follows from the known inequalities.

7. Representing problems as graphs. We say a matrix has dimensions $m \times n$ if it has m rows and n columns. If matrix A has dimensions $x \times y$ and matrix B has dimensions $z \times w$, then the product AB exists if and only if $y = z$. In the case where the product exists, AB will have dimensions $x \times w$. In this problem, we are given a list of matrices and their dimensions, and we want to determine if there is an order in which we can multiply all the matrices together, using each matrix exactly once. The list of matrices and their dimensions is as follows:

A is 4×5,
B is 5×7,
C is 5×4,
D is 4×7,
E is 7×5.

(a) Draw a graph that represents this situation in such a way that finding an order we seek corresponds to finding a Hamilton Tour of your graph. Describe what the vertices of your graph represent, and when two vertices are connected with an edge.

(b) Draw a different graph that represents this situation in such a way that finding an order we seek corresponds to to finding an Euler Tour of your graph. Describe what the vertices of your graph represent, and when two vertices are connected with an edge.

(c) Give an order in which we can multiply these matrices, or say that no such order exists.

8. Eulerian and Hamiltonian path/circuit. Han and Chewie are on the run from the Imperial Army. Starting from the Rebel's base on Yavin 4, in order to shake off Darth Vader's pursuit, they have to make a series of hyperspace jumps to seven different planets then return to the original base. However, the Millennium Falcon cannot make any jump longer than 12 *parsec* (Note: *parsec* is a unit of distance, not time). They looked up their star map and made the following table of the distances between planets. All the entries below are measured in *parsec*.

	Yavin 4	Coruscant	Tatooine	Hoth	Endor	Naboo	Kessel	Dagobah
Yavin 4	0	10	15	15	18	14	16	12
Coruscant	10	0	12	13	16	12	14	10
Tatooine	15	12	0	10	12	9	10	12
Hoth	15	13	10	0	12	12	13	15
Endor	18	16	12	12	0	12	11	16
Naboo	14	12	9	12	12	0	9	12
Kessel	16	14	10	13	11	9	0	14
Dagobah	12	10	12	12	16	12	14	0

(a) Draw a graph that will help Han and Chewie plan their escape. Specify whether your graph is directed or undirected. Carefully describe what the vertices of your graph represent, and when two vertices are connected with an edge.

(b) Say which graph theory problem they are trying to solve on this graph.

(c) Find a feasible escape route for Han and Chewie that meets the constraints, or explain why no such route exists.

9. DAGs A daily flight schedule is a list of all the flights taking place that day. In a daily flight schedule, each flight F_i has an origin city OC_i, a destination city DC_i, a departure time d_i and an arrival time $a_i > d_i$. This is an example of a daily flight schedule for June 08, 2019 (our exam date), listing flights as $F_i = (OC_i, DC_i, d_i, a_i)$:

$$F_1 = \text{(Portland, Los Angeles, 7:00am, 9:00am)}$$
$$F_2 = \text{(Portland, Seattle, 8:00am, 9:00am)}$$
$$F_3 = \text{(Los Angeles, San Francisco, 8:00am, 9:30am)}$$
$$F_4 = \text{(Seattle, Los Angeles, 9:30am, 11:30am)}$$
$$F_5 = \text{(Los Angeles, San Francisco, 12:00pm, 1:00pm)}$$
$$F_6 = \text{(San Francisco, Portland, 1:30pm, 3:00pm)}$$

(a) Given any daily flight schedule, describe how to construct a DAG so that paths in the DAG represent possible sequences of connecting flights a person could take. What are the vertices, and when are two vertices connected with an edge?

(b) Why is your graph always a DAG?

(c) What problem would you need to solve in your DAG to help you determine the maximum number of flights a person could take on a given day?

(d) Draw the DAG you described for the given example of June 08, 2019 and give the maximum number of flights a person could take on that day.

10. Trees. A binomial tree is a special kind of rooted tree used for various data structures in computer science. A degree d binomial tree can be defined recursively as follows. A degree 0 binomial tree is a single vertex with no edges. A degree d binomial tree has a root vertex with out-degree d. The first (that is, leftmost) subtree is a degree $d-1$ binomial tree. The second (that is, second to left) subtree is a degree $d-2$ binomial tree. Continue on in this way so that the last (rightmost) subtree is a degree 0 binomial tree.

(a) What is the height of a degree d binomial tree? Prove your result by induction on d.

(b) Write a recurrence for the number of nodes $N(d)$ in a binomial tree of degree d.

(c) Use the guess-and-check method to guess a formula for $N(d)$. Prove that your formula holds by induction on d.

11. Stars and Bars item Let n, k be positive integers. Recall the *Stars and Bars Problem* that the integer equations $a_1 + a_2 + \ldots + a_k = n$, where $a_i \geq 0$, has $\binom{n+k-1}{k-1}$ solutions.

Find the number of integer solutions of the equation $x_1 + x_2 + x_3 = 50$ such that $0 \leq x_i \leq 19$, for $i = 1, 2, 3$. You may leave your answer in terms of the binomial coefficients.

(*Hint: Let A_i be the set of all solutions in which $x_i \geq 20$.*)

1. Order questions. For each of the following pairs of functions f and g, choose whether $f \in o(g)$, $f \in \Theta(g)$, or $f \in \omega(g)$ (exactly one will be true).

base is not specified

(a) $f(n) = n, g(n) = 2^{\lfloor \log n \rfloor}$ $f \in \omega(g)$ $f \in \Theta(g)$ limit of their ratio

(b) $f(n) = \log(n^4), g(n) = (\log n)^4$. $f \in o(g)$

(c) $f(n) = n^2, g(n) = 1000n^2 + 2^{100}$ $f \in \Theta(g)$

(d) $f(n) = 2^{2n}, g(n) = 2^n$. $f \in \omega(g)$

(e) $f(n) = n!, g(n) = n^n$. $f \in o(g)$

(a) $\log n - 1 < \lfloor \log n \rfloor \leq \log n$ 4^n 2^n

$$\lim_{n\to\infty} \frac{f(n)}{g(n)}$$ f g

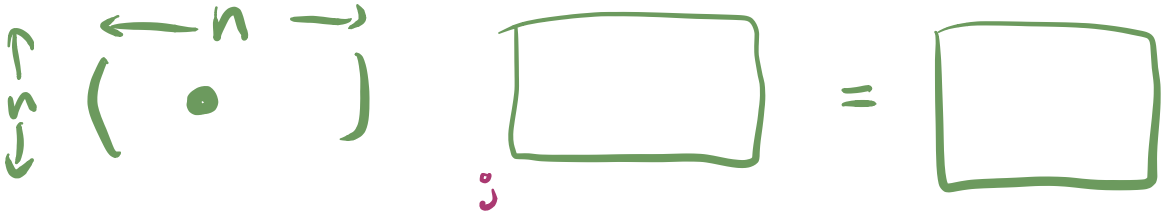
$f \in o(g)$

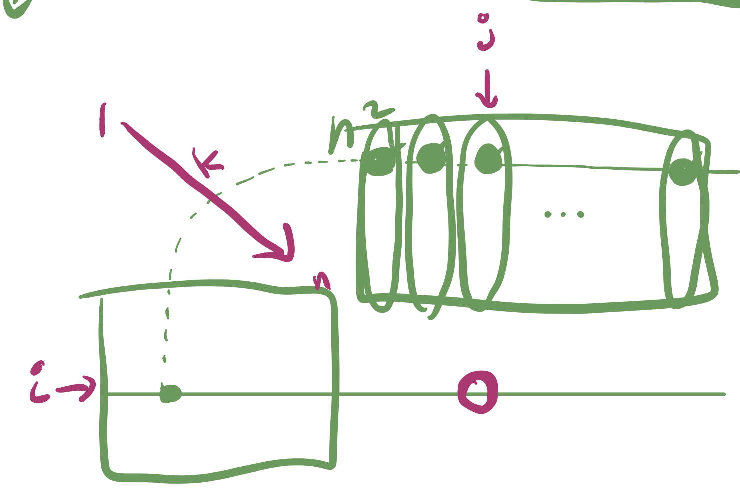
(b) $$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{\log(n^4)}{(\log n)^4} = 4 \lim_{n\to\infty} \frac{\cancel{\log n}}{(\log n)^{\cancel{4}\,3}} = 0$$

$f \in o(g)$

(c)

2. Analyzing algorithms. What is the order of the standard multiplication algorithm for multiplying two $n \times n$ matrices? Suppose that any two matrix entries can be multiplied together in constant time, and that addition takes constant time.





for an entry in left matrix, there should be n times of multiplication.

there are n^2 entries. So, the order of standard multiplication algorithm is $O(n^3)$

a, b → input matrices of $n \times n$?

okay !

```
n → for i = 1 to n
  n → for j = 1 to n
    n → for k = 1 to n
      c → c[i][j] = a[i][k] * b[k][j];
```

☺(n^3)

3. Recursive algorithms. Here's a recursive algorithm, that given an n bit number in binary, $x = x_{n-1} \ldots x_0$ and an $m < n$ bit number in binary y, computes $x \mod y$. It uses the following sub-routines: $Add(x, y)$ adds two n bit binary numbers in $O(n)$ time, $AsLarge(x, y)$ returns true if and only if $x \geq y$ and also takes $O(n)$ time, and $Subtract(x, y)$ computes $x - y$ and is also $O(n)$ time. We assume $y \geq 2$ and $n \geq 1$.

Mod $(x = x_{n-1} \ldots x_1 x_0,\ y)$ $\quad T(n+1) = T(n) + O(n)$ $\quad$ an algorithm

1. IF $n = 1$, return x_0.
2. $v \leftarrow Mod(x_{n-1} \ldots x_1, y)$. $\quad v$
3. $v \leftarrow Add(v, v)$ $\rightarrow O(m)$
4. $v \leftarrow Add(v, x_0)$. $\rightarrow O(m)$
5. IF $AsLarge(v, y)$ THEN $v \leftarrow Subtract(v, y)$. $\quad O(n) \quad O(m)$
6. Return v

$2 \mid 13$

(a) **Correctness.** Prove by induction on n that $Mod(x, y)$ returns $x \mod y$.

$2 \mid 6 \ldots 1$ Hints:

- Let $x = 2a + x_0$, where $a = x_{n-1} \ldots x_1$. $\quad x \bmod y = (2a + x_0) \bmod y = 2$

$2 \mid 3 \ldots 0$ - To show that two numbers are the same mod y, show that their difference is a multiple

$1 \mid 1 \ldots 1$ of y.

0 (b) **Recurrence.** Give a recurrence for $T(n)$, the time this algorithm takes on an input of size n.

$T(n+1) = T(n) + O(m)$

(c) **Solving recurrences.** Solve this recurrence to give the Big O time of this algorithm.

(a) Mod

Base Case: when $n=1$, by 1, x_0 is returned

Since $x_0 < 2 \leq b$, $x_0 = x \bmod y$. So, the algorithm is true for $n=1$.

Induction hypothesis: for some integer k, the algorithm ~~holds~~ works when $n=k$. then on Line 2, v is set to $a \bmod y$ with $x = 2a + x_0$.

after line 3, v is set to $2 \cdot (a \bmod y)$ by the I.H

4 $\quad v$ is set to $2(a \bmod y) + x_0$

Since $a \bmod y \leq y-1$, $2(y-1) + 1 \geq 2(a \bmod y) + x_0$

$2y - 1 \geq 2(a \bmod y) + x_0$

after 5, $0 \le v \le y-1$. and $v \equiv 2(a \bmod y) + x_0 \bmod y$

$$\left(2(a \bmod y) + x_0\right) - \left(\quad x \quad \right)$$

$$= 2(a \bmod y + \cancel{x_0}) - (2a + \cancel{x_0})$$

$$= 2(a \bmod y - a)$$

(By the division theorem $\exists q \in \mathbb{Z}$ with $a = qy + (a \bmod y)$. So,
$a \bmod y - a = -qy$)

$= -2qy$. So, v is equal to the remainder of x. therefore, by the induction on n,

the algorithm is sound.

(b) since $a \bmod y$ cannot be bigger than the direct y, it's $O(m)$

$\therefore T(n+1) = T(n) + O(m)$

(c) By unraveling,

$$\begin{aligned} T(n) &= \cancel{T(n-1)} + O(m) \\ \cancel{T(n-1)} &= \cancel{T(n-2)} + O(m) \\ + \quad \cancel{T(1)} &= T(0) + O(m) \end{aligned}$$

$$T(n) = T(0) + n\,O(m)$$

$$\therefore \quad T(n) \in O(nm)$$

5. Recursive Counting

For problems (a)-(e), it is not required to solve the recurrence:

(a) In a round-robin tennis tournament with n players, every tennis player plays against every other player. Let $T(n)$ be the total number of tennis matches taking place among the n players. Find a recurrence for $T(n)$ and explain in words why $T(n)$ satisfies this recurrence.

$T(1) = 0$, $T(n+1) = T(n) + n$ $(\forall n \geq 1)$

n

(b) In a board game, players must place colored tiles in a single row. There are red 1×1 tiles, yellow 2×1 tiles, blue 2×1 tiles, purple 3×1 tiles, and green 3×1 tiles. Let $C(n)$ be the number of different $n \times 1$ colored ~~rows~~ coloumn that can be created using these tiles. Find a recurrence for $C(n)$ and explain in words why $C(n)$ satisfies this recurrence.

(c) Any binary string can be broken into contiguous chunks of the same character, called runs. For example, 00111010000 has:

- a run of 0s of length 2
- a run of 1s of length 3
- a run of 0s of length 1
- a run of 1s of length 1
- a run of 0s of length 5

11 0

Let $B(n)$ be the number of length n binary strings where each run of 1s has even length. Find a recurrence for $B(n)$ and explain in words why $B(n)$ satisfies this recurrence.

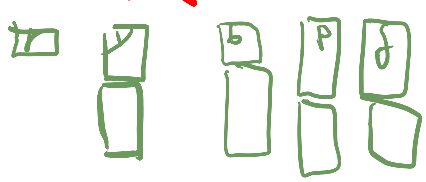
(d) Let $S(n)$ be the number of subsets of $\{1, 2, \ldots, n\}$ having the following property: there are no three elements in the subset that are consecutive integers. Find a recurrence for $S(n)$ and explain in words why $S(n)$ satisfies this recurrence.

(b) $C(0) = \cancel{0}$ $C(1) = 1$ $C(2) = 3$, $C(3) = C(2) + 2C(1) + 2C(0)$

$= 3 + 2 + 2$

$= 7$

r
r y b



$C(n) = C(n-1) + 2C(n-2) + 2C(n-3)$

starting with red $\to C(n-1)$

yellow $\to C(n-2)$

blue $\longrightarrow C(n-2)$

purple $\longrightarrow C(n-3)$

green $\longrightarrow C(n-3)$

(c) $B(0)=1$ $B(1)=1$ the very front of our sequences

starts with 0 or 1

① ②

if it starts with 1, there should not be 0 in the second place.

① 0 $\boxed{B(n-1)}$

② 1 1 $\boxed{B(n-2)}$

By the sum rule,

$$B(n) = B(n-1) + B(n-2), \quad B(0)=B(1)=1.$$

(d) Basecase $S(0) = 2^{|\emptyset|} = 1$

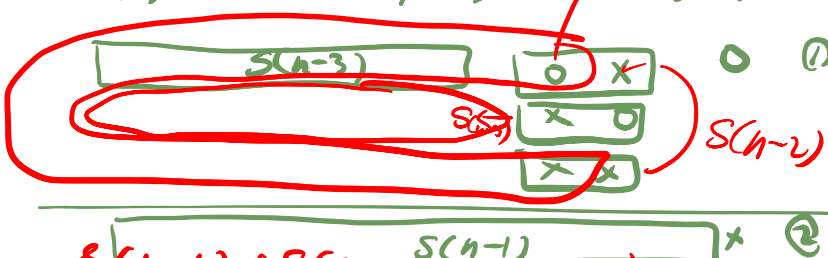
× $S(1) = 2^1 = 2$

× $S(2) = 2^2 = 4$

×

1, 2, 3, $\cdots$, n-3, n-2, n-1, n

~~You~~ I cannot explain that one. sorry. I was wrong.



$S(n-1)+S(n-2)+S(n-3)$

$$S(n) = S(n-1) + \cancel{3S(n-3)}, \quad S(0)=1, S(1)=2, S(2)=4$$

$(n \geq 3)$

4. Counting. (a) How many 5-card hands can be formed from an ordinary deck of 52 cards if exactly two suits are present in the hand?

(b) In any bit string, the longest consecutive run length is the maximum number of consecutive 1's or consecutive 0's in the string. For example, in the string 1101000111, the longest consecutive run length is 3. How many bit strings of length 10 have a longest consecutive run length of 6?

(c) A software company assigns its summer interns to one of three divisions: design, implementation, and testing. In how many ways can a group of ten interns be assigned to these divisions if each division needs at least one intern?

(d) How many numbers in the interval [1,10000] are divisible by 7, 9, or 11?

(e) A California license plate follows the format $DLLLDDD$, where D represents a digit and L represents an upper case letter. For example, a valid California license plate is $5JBC434$ (Quang's plate). How many California license plates are possible? $26^3 10^4$

Under a fixed-length encoding scheme, how many bits of memory does the California DMV need to allocate in its database to store each license plate number?

$\lceil \log_2 26^3 10^4 \rceil$

$\lceil 27.3 \rceil = 28$

(a)

$\binom{4}{2}$ Choosing the 2 suits.

$\binom{4}{1}\binom{13}{4}\binom{3}{1}\binom{13}{1}$

$+$

$\binom{4}{1}\binom{13}{3}\binom{3}{1}\binom{13}{2}$

heart spade

13 13

All hearts All spades

$\binom{26}{5} - \left(\binom{13}{5} + \binom{13}{5}\right)$

$\therefore \binom{4}{2}\left(\binom{26}{5} - 2\binom{13}{5}\right)$

(b) determine if it is 0 or 1. → 2 ways.

000000 0 1

— — — — —

choose where to put the consecutive #'s.

_ _ _ 1 $\boxed{000000}$

edge? 1 $\boxed{}$

left or right · rest

$2 * 2^3$

middle?

middles · rest

$3 * 2^2$

1 $\boxed{}$ 1 _ _

by the sum rule and product rule,

$$2(2 \cdot 2^3 + 3 \cdot 2^2)$$

$$= 2(16 + 12) = \boxed{56.}$$

(c)

student1
student2
⋮
student10

div1
div2
div3

Let A be the set of $(\overbrace{1, 2, 3, \ldots, 1, \ldots}^{10})$ where 1 is in it

B 2

C 3

$$|A \cap B \cap C| = |U| - |A^c \cup B^c \cup C^c|$$

$$|A^c \cup B^c \cup C^c| = |A^c| + |B^c| + |C^c| - |A^c \cap B^c| - |B^c \cap C^c| - |C^c \cap A^c| + |A^c \cap B^c \cap C^c|.$$

$$= 2^{10} + 2^{10} + 2^{10} - 1^{10} - 1^{10} - 1^{10} + 0^{10}$$

$$= 3(2^{10} - 1) = 3 \cdot 1023$$

$$= 3069$$

$$|A \cap B \cap C| = \boxed{3^{10} - 3069}$$

(d)

Let D_7 the set of numbers in $[1, 10000]$ that are divisible by 7.

Let D_9 " 9

D_{11} " 11.

$$|D_7 \cup D_9 \cup D_{11}| = \underbrace{|D_7|}_{\lfloor \frac{10k}{7} \rfloor} + \underbrace{|D_9|}_{\lfloor \frac{10k}{9} \rfloor} + \underbrace{|D_{11}|}_{\lfloor \frac{10k}{11} \rfloor} - \underbrace{|D_7 \cap D_9|}_{\lfloor \frac{10k}{63} \rfloor} - \underbrace{|D_9 \cap D_{11}|}_{\lfloor \frac{10k}{99} \rfloor} - \underbrace{|D_{11} \cap D_7|}_{\lfloor \frac{10k}{77} \rfloor} + \underbrace{|D_7 \cap D_9 \cap D_{11}|}_{\lfloor \frac{10k}{693} \rfloor}$$

(e) $26^3 \times 10^4$, $\boxed{28}$

5. Probability. (a) I have 10 shirts, 6 pairs of pants, and 3 jackets. Every day I dress at random, picking one of each category. What is the probability that today I am wearing at least one garment I was wearing yesterday?

(b) A permutation of size n is a rearrangement of the numbers $\{1, 2, \ldots, n\}$ in any order. A *rise* in a permutation occurs when a larger number immediately follows a smaller one. For example, if $n = 5$, the permutation 1 3 2 4 5 has three rises. What is the expected number of rises in a permutation of size n?

(c) There are n teams in a sports league. Over the course of a season, each team plays every other team exactly once. If the outcome of each game is a fair random coin flip, and there are no ties, what is the probability that some team wins all of its games?

(d) Suppose there are 10 people of distinct heights standing in line for free food, arranged in a random order. Use linearity of expectation to calculate the expected number of people who can see all the way to the front of the line.

(a) 10 6 3

$1 - 1 \times \cancel{\frac{9}{10} \cdot \frac{5}{6} \cdot \frac{2}{3}} = \boxed{\frac{1}{\cancel{180}\ 2}}$

2

(b) Rise and Fall

for each arrangement, the

rise + fall = n−1

$E(\text{rise}) + E(\text{fall}) = n-1$

$E(\text{rise}) = E(\text{fall})$ since the arrangement is random

$E(\text{rise}) = \boxed{\frac{n-1}{2}}$ ✓

$X_i = 1$ if rise at position i

$X_i = 0$ false at i

$E(X_i) = 1 \cdot \frac{1}{2} = \frac{1}{2}$

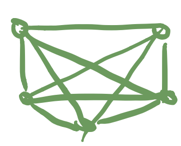
$X = \sum_{i=1}^{n-1} X_i$

$E(X) = \sum_{i=1}^{n-1} E(X_i) = \frac{n-1}{2}$

(c) a team and b team $\binom{n}{2}$ $\binom{n-1}{2}$

a wins $\to \frac{1}{2}$ $\binom{n}{2}$

b wins $\to \frac{1}{2}$ $2^{\binom{n-1}{2}}$



X_i : team i

$X_i = 1$ team i won all the games?

$X_i = 0$ team i lose

$P(X_i = 1) = \frac{1}{2^{n-1}}$

$E(X) = \frac{n}{2^{n-1}}$

(e) Assume that every time you go on a job interview, your chance of getting a job offer is 25%. How many job interviews must you go on so that the probability of your getting a job offer is greater than 95%?

(f) Suppose that you roll two dice and don't get to look at the outcome. Your friend looks at the outcome and tells you honestly that at least one of the dice came up 6. What is the probability that the sum of your two dice is 8?

(g) Suppose there are n people assigned to m different tasks. Assume that each person is randomly assigned a task and that for each person, all tasks are equally likely. Use linearity of expectation to find the expected number of people working on a task alone.

(h) Tesla is planning to introduce a new self-driving car. The company commissions a marketing report for each new car that predicts either the success or the failure of the car. Of the new cars introduced by the company so far, 65% have been successes. Furthermore, 70% of their successful cars were predicted to be successes, while 40% of failed cars were predicted to be successes. Find the probability that this new self-driving car will be successful if its success has been predicted.

oh yeah

(i) Say we have a coin that has probability p of landing on Heads, and probability $1 - p$ of landing on Tails. Let X_n be the random variable that indicates the number of heads in n independent tosses of this coin. What is $\boxed{\text{the variance of } X_1}$? What is $\boxed{\text{the variance of } X_n}$?

(d)

4 2 1 10 9 6 3 8 7 5

x x x o o x x o o o

7 8 5

front

7 5 8

8 7 5

5 7 8

8 5 7

5 8 7

X_i: Can i^{th} person from front see all the back?

$E(X_i) = 1\,P(X_i = 1) + 0\cdot(1-P(X_i=1))$

$= \frac{1}{i}$

$E(X) = \sum_{i=1}^{n} E(X_i) = \boxed{\sum_{i=1}^{n} \frac{1}{i}} = \cancel{H_n}$

(e)

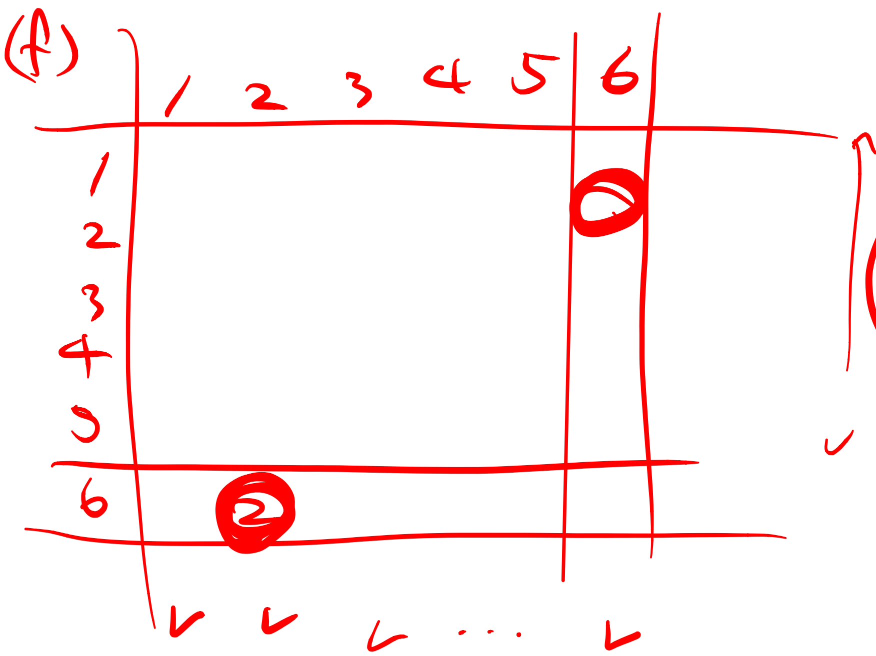
yes job $\rightarrow \frac{1}{4}$

no jobs $\rightarrow \frac{3}{4}$

$$1 - P(\underbrace{\text{not getting an offer after } i \text{ tries}}_{\geq .95})$$

$$.05 \geq \left(\frac{3}{4}\right)^i$$

(f)



$\frac{2}{11}$

6

(g)

$X_i = 1$ if ith person works alone.

$\quad\;\; 0$ otherwise.

n people m tasks

$$P(X_i = 1) = \frac{\overset{i\text{th}}{m} \cdot (\overset{\text{rest}}{m-1})^{n-1}}{m^n} = \left(\frac{m-1}{m}\right)^{n-1}$$

i

$n-1$ $m-1$ $(m-1)^{n-1}$

$\frac{m-1}{m}$

$$E(X_i) = \left(\frac{m-1}{m}\right)^{n-1}$$

(h)

	predict success	not p	
Success	$65 \times .7$	$65 \times .3$	65
no success	$35 \times .4$	$35 \times .6$	35
			100

$$\boxed{\frac{65 \times .7}{65 \times .7 + 35 \times .4}}$$

(i)

X^2	0	1
X	0	1
	$\frac{1}{2}$	$\frac{1}{2}$

$E(X^2) = \frac{1}{2}$ $\quad E(X) = \frac{1}{2}$ $\quad V(X) = \frac{1}{2} - \left(\frac{1}{2}\right)^2 = \boxed{\frac{1}{4}}$

X_n^2	0	1^2		i^2	n^2
X_n	0	1	$\cdots$	i	$\cdots$ n
$P(X_n=x)$	$\frac{1}{2^n}$	$\frac{{}_nC_1}{2^n}$	$\cdots$	$\frac{{}_nC_i}{2^n}$	$\cdots$ $\frac{1}{2^n}$

$$\boxed{V(X_n)} = E(X_n^2) - (E(X_n))^2 = \boxed{\sum_{i=0}^{n} i^2 \frac{{}_nC_i}{2^n} - \left(\sum_{i=0}^{n} \frac{i\,{}_nC_i}{2^n}\right)^2}$$

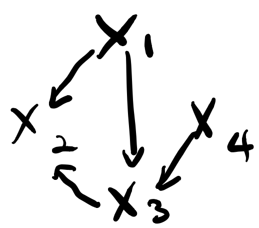
6. Representing problems as graphs. You have a system of n variables representing real numbers, $X_1, \ldots, X_n$. You are given a list of inequalities of the form $X_i < X_j$ for some pairs i and j. You want to know whether you can deduce with certainty from the given information that $X_1 < X_n$.

For example, say n is 4. A possible input is the list of inequalities $X_1 < X_2, X_1 < X_3, X_4 < X_3$ and $X_3 < X_2$. Does it follow that $X_1 < X_4$?

(a) Give a description of a directed graph that would help solve this problem. Be sure to define both the vertices and edges in terms of the variables and known inequalities.

(b) Draw the graph you described for the example above. Does $X_1 < X_4$ follow? Why or why not?

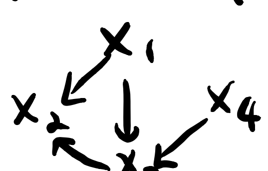
(c) Say which algorithm from lecture we could use on such a graph to determine whether $X_1 < X_n$ follows from the known inequalities.



(a) Vertices are numbers X_i , $i \in \mathbb{Z}$ for $1 \le i \le n$.

For each of the inequality in the list, Directed edges are defined from the vertex that has less numbers. to the larger number. If there is a path from X_1 to X_n then we can deduce that $X_1 < X_n$.

(b)



there is no path from X_1 to X_4. So, We cannot deduce from the list that $X_1 < X_4$.

(c) ? Depth first search! or Breath first search to find if there is a path from X_1 to X_n.

7. Representing problems as graphs. We say a matrix has dimensions $m \times n$ if it has m rows and n columns. If matrix A has dimensions $x \times y$ and matrix B has dimensions $z \times w$, then the product AB exists if and only if $y = z$. In the case where the product exists, AB will have dimensions $x \times w$. In this problem, we are given a list of matrices and their dimensions, and we want to determine if there is an order in which we can multiply all the matrices together, using each matrix exactly once. The list of matrices and their dimensions is as follows:

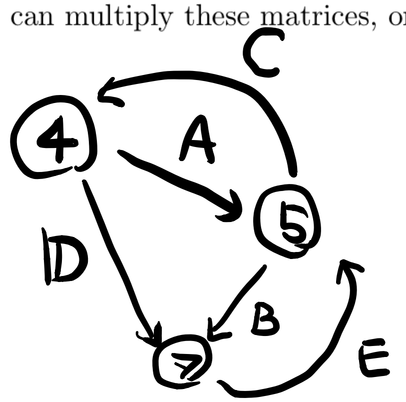
A is 4×5,
B is 5×7,
C is 5×4,
D is 4×7,
E is 7×5.

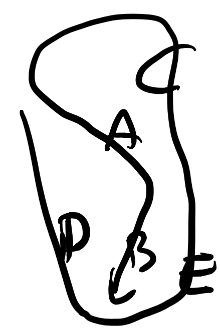
(a) Draw a graph that represents this situation in such a way that finding an order we seek corresponds to finding a Hamilton Tour of your graph. Describe what the vertices of your graph represent, and when two vertices are connected with an edge.

(b) Draw a different graph that represents this situation in such a way that finding an order we seek corresponds to to finding an Euler Tour of your graph. Describe what the vertices of your graph represent, and when two vertices are connected with an edge.

(c) Give an order in which we can multiply these matrices, or say that no such order exists.

(a)





Edges are representative of a matirices.
vertices are # of raws or # of columns.
Directions are given in the rule that from the # of raws to # of columns.

(b) DECAB , ABECD , ACDEB

8. Eulerian and Hamiltonian path/circuit. Han and Chewie are on the run from the Imperial Army. Starting from the Rebel's base on Yavin 4, in order to shake off Darth Vader's pursuit, they have to make a series of hyperspace jumps to seven different planets then return to the original base. However, the Millennium Falcon cannot make any jump longer than 12 *parsec* (Note: *parsec* is a unit of distance, not time). They looked up their star map and made the following table of the distances between planets. All the entries below are measured in *parsec*.

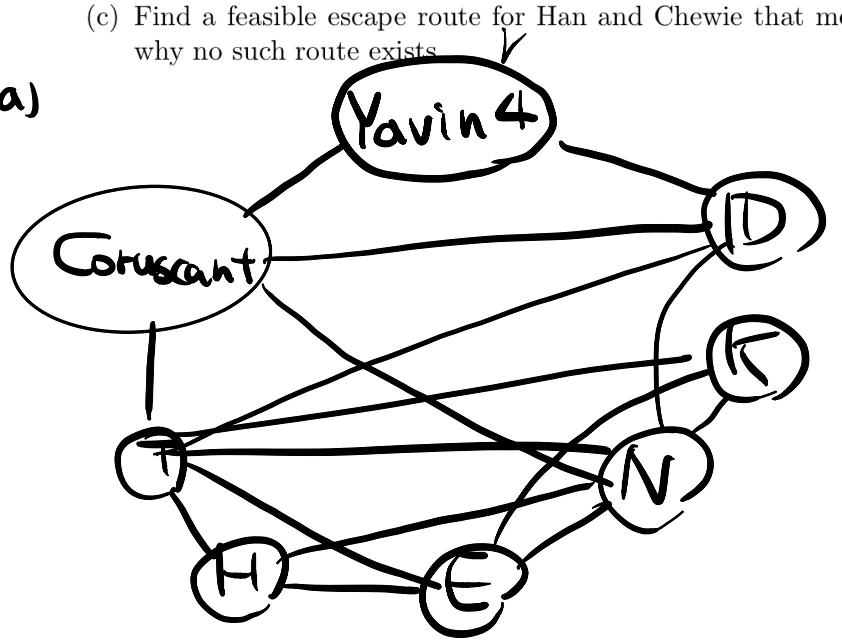
	Yavin 4	Coruscant	Tatooine	Hoth	Endor	Naboo	Kessel	Dagobah
1 Yavin 4	✓	10 C	~~15~~	~~15~~	~~18~~	~~14~~	~~16~~	12 D
2 Coruscant	10	✓	12 T	~~13~~	~~16~~	12 N	~~14~~	10 D
3 Tatooine	~~15~~	12	✓	10 H	12 E	9 N	10 K	12 D
4 Hoth	~~15~~	~~13~~	10	✓	12 E	12 N	~~18~~	~~15~~
5 Endor	~~18~~	~~16~~	12	12	✓	12 N	11 K	~~16~~
6 Naboo	~~14~~	12	9	12	12	✓	9 K	12 D
7 Kessel	~~16~~	~~14~~	10	~~13~~	11	9	✓	~~14~~
8 Dagobah	12	10	12	12 ?	~~16~~	12	~~14~~	✓

(a) Draw a graph that will help Han and Chewie plan their escape. Specify whether your graph is directed or undirected. Carefully describe what the vertices of your graph represent, and when two vertices are connected with an edge.

(b) Say which graph theory problem they are trying to solve on this graph.

(c) Find a feasible escape route for Han and Chewie that meets the constraints, or explain why no such route exists.

(a)



Cycle 찾기

, Directed no cycle

9. DAGs A daily flight schedule is a list of all the flights taking place that day. In a daily flight schedule, each flight F_i has an origin city OC_i, a destination city DC_i, a departure time d_i and an arrival time $a_i > d_i$. This is an example of a daily flight schedule for June 08, 2019 (our exam date), listing flights as $F_i = (OC_i, DC_i, d_i, a_i)$:

$OC_i \xrightarrow{F_i} DC_i$

d_i a_i

$F_1 =$ (Portland, Los Angeles, 7:00am, 9:00am)

$F_2 =$ (Portland, Seattle, 8:00am, 9:00am)

$F_3 =$ (Los Angeles, San Francisco, 8:00am, 9:30am)

$F_4 =$ (Seattle, Los Angeles, 9:30am, 11:30am)

$F_5 =$ (Los Angeles, San Francisco, 12:00pm, 1:00pm)

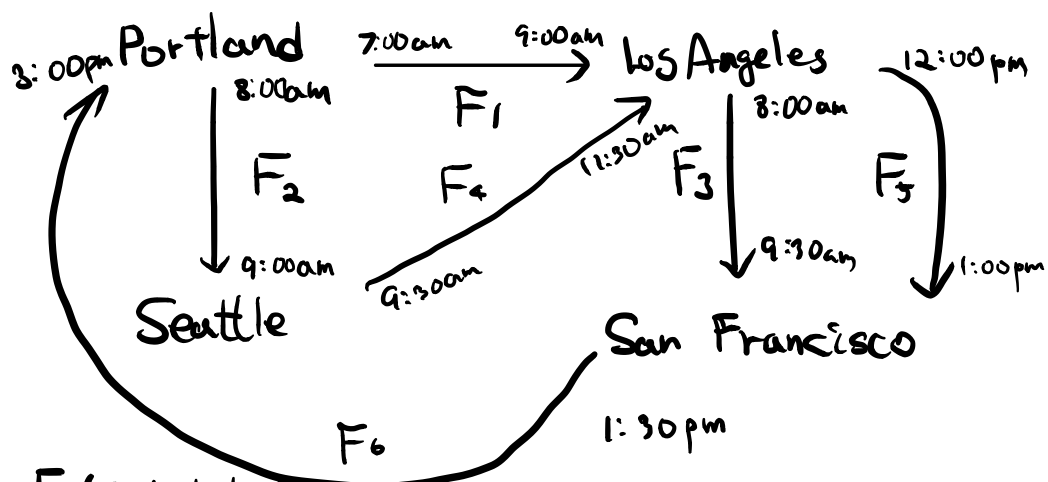
$F_6 =$ (San Francisco, Portland, 1:30pm, 3:00pm)

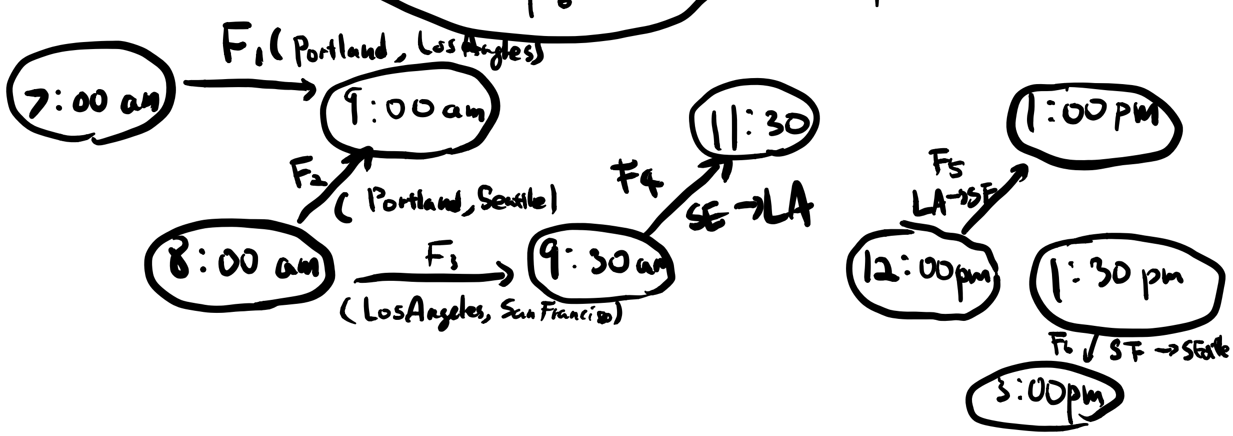
(a) Given any daily flight schedule, describe how to construct a DAG so that paths in the DAG represent possible sequences of connecting flights a person could take. What are the vertices, and when are two vertices connected with an edge?

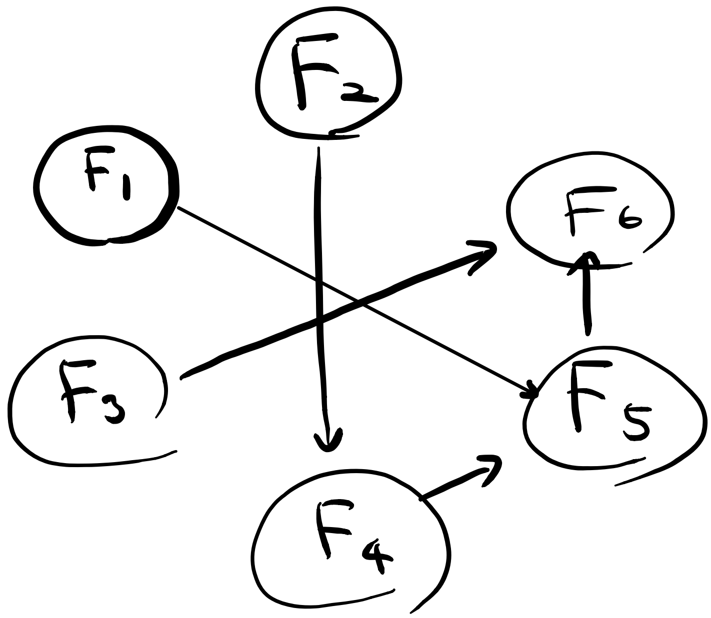
(b) Why is your graph always a DAG? we cannot take a flight twice a day.

(c) What problem would you need to solve in your DAG to help you determine the maximum number of flights a person could take on a given day? the longest path

(d) Draw the DAG you described for the given example of June 08, 2019 and give the maximum number of flights a person could take on that day.







F_1	F_5
F_2	F_4
F_3	F_6
F_4	F_5
F_5	F_6
F_6	

$F_2 \to F_4 \to F_5 \to F_6$

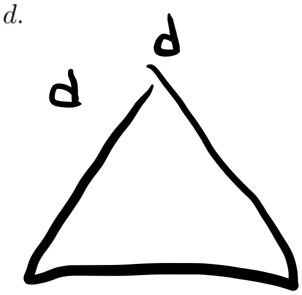
4 flights are possible.

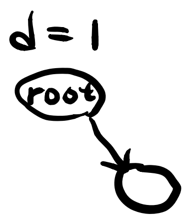
10. Trees. A binomial tree is a special kind of rooted tree used for various data structures in computer science. A degree d binomial tree can be defined recursively as follows. A degree 0 binomial tree is a single vertex with no edges. A degree $\cancel{d}$ binomial tree has a root vertex with out-degree $\cancel{d}$. The first (that is, leftmost) subtree is a degree $d-1$ binomial tree. The second (that is, second to left) subtree is a degree $d-2$ binomial tree. Continue on in this way so that the last (rightmost) subtree is a degree 0 binomial tree.

(a) What is the height of a degree d binomial tree? Prove your result by induction on d.

(b) Write a recurrence for the number of nodes $N(d)$ in a binomial tree of degree d.

(c) Use the guess-and-check method to guess a formula for $N(d)$. Prove that your formula holds by induction on d.

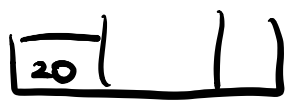




11. Stars and Bars item Let n, k be positive integers. Recall the *Stars and Bars Problem* that the integer equations $a_1 + a_2 + \ldots + a_k = n$, where $a_i \geq 0$, has $\binom{n+k-1}{k-1}$ solutions.

Find the number of integer solutions of the equation $x_1 + x_2 + x_3 = 50$ such that $0 \leq x_i \leq 19$, for $i = 1, 2, 3$. You may leave your answer in terms of the binomial coefficients.

(*Hint: Let A_i be the set of all solutions in which $x_i \geq 20$.*)

A_1  30

A_2  30

A_3  30

$$|A_1^c \cap A_2^c \cap A_3^c| = |U| - |A_1 \cup A_2 \cup A_3| = |U| - \left|\bigcup_{i=1}^{3} A_i\right|$$

$$= |U| - |A_1| - |A_2| - |A_3| + |A_1 \cap A_2| + |A_2 \cap A_3| + |A_3 \cap A_1| - \underbrace{|A_1 \cap A_2 \cap A_3|}_{=0}$$

$$= |U| - 3|A_1| + 3|A_1 \cap A_2| - \cancel{|A_1 \cap A_2 \cap A_3|}$$

$$= {}_3H_{50} - 3\,{}_3H_{30} + 3\,{}_3H_{10}$$

$$= \boxed{\binom{52}{2} - 3\binom{32}{2} + 3\binom{12}{2}}$$

CSE 21 – Spring 2019 – Practice Final Solution

1. Order questions. For each of the following pairs of functions f and g, choose whether $f \in o(g), f \in \Theta(g)$, or $f \in \omega(g)$ (exactly one will be true).

(a) $f(n) = n, g(n) = 2^{\lfloor \log n \rfloor}$

(b) $f(n) = \log(n^4), g(n) = (\log n)^4$.

(c) $f(n) = n^2, g(n) = 1000n^2 + 2^{100}$

(d) $f(n) = 2^{2n}, g(n) = 2^n$.

(e) $f(n) = n!, g(n) = n^n$.

Solution:

(a) First, observe that $n = 2^{\log n}$. So, $2^{\lfloor \log n \rfloor} < 2^{\log n} = n$. This suggests the Θ relationship for these functions, because they are always very close to each other. Let's pick constants to prove this:

$$c_1 \times 2^{\lfloor \log n \rfloor} \leq n \leq c_2 \times 2^{\lfloor \log n \rfloor}$$

By the above, we can set $c_1 = 1$, and $c_2 = 2$. , making the rhs of the above inequality $2^{\lfloor \log n \rfloor + 1}$. This (more than) makes up for the fractional power lost to flooring, and proves $f \in \Theta(g)$.

(b) Again, we first simplify the functions and consider their relationship intuitively. Using properties of logs, $f(n) = \log(n^4) = 4\log(n)$. Since $g = \log(n)^4$, this suggests that eventually $g(n) > f(n)$. We'll use the limit test to confirm this:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{4\log(n)}{\log(n)^4} = \frac{4}{\log(n)^3} = 0$$

Proving that $f \in o(g)$.

(c) Considering the functions informally, we see that the n^2 term will dominate both functions asymptotically. This suggests that they are equivalent, and we should attempt to prove the Θ relationship.

One direction and constant is clear: $n^2 < 1000 \times n + 2^{100}$, so the constant is 1. In the other direction, let's set up and inspect the desided inequality:

$$c(1000n^2 + 2^100) < n^2$$
$$c1000n^2 + c2^100 < n^2$$

Setting $c = 2^{-100}$ and denoting $c \times 1000 < 1$ by ϵ because it is a very small number that is guranteed to be less than 1 with that setting of c:

$$\epsilon n^2 + 1 < n^2$$

Since $\epsilon < 1$, the inequality above holds for any n sufficiently large. We can set $n_0 = 4$, proving $f \in \Theta(g)$.

(d) Comparing the functions informally, we have $2^{2n} > 2^n$ for any n. So this suggests a ω relationship, and the following limit test:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \frac{2^{2n}}{2^n} = 2^n = \infty$$

proving $f \in \omega(g)$.

(e) Unrolling the definitions, we know that $n! < n^n$ for any n, so this suggests the o relationship. The limit is annoying to evaluate, so we use the definitions directly. The desired inequality is, for *any* c, eventually

$$cn! < n^n$$

This is true because we can divide both sides by $n!$, and eventually $n^n/n! = \frac{n}{n} \cdot \frac{n}{n-1} \cdots \frac{n}{1} > c$ for any c.

2. Analyzing algorithms. What is the order of the standard multiplication algorithm for multiplying two $n \times n$ matrices? Suppose that any two matrix entries can be multiplied together in constant time, and that addition takes constant time.

Solution: To analyze the standard matrix multiplication algorithm, we must first know the standard matrix multiplication algorithms. Let A and B be $n \times n$ matricies. The standard algorithm is:

1. For each i in $1 \dots n$
   (a) For each j in $1 \dots n$
      i. do: $[AB]_{i,j} = \sum_{r=1}^{n} A_{i,r} \times B_{r,j}$

Note that we were told that, in this situation, addition and multiplication are constant-time. Looking at the loop structure, we see that the inner-most line of code (line i) is executed once for each pair (i, j). Since the matricies are of dimension $n \times n$, this inner line must be executed n^2 times. So we can obtain the full cost of the algorithm by multiplying the cost of the inner line by n^2 (number of times it is executed).

The summation in the inner line runs from 1 to n, so there are n addition operations. Each addition operation is run on the output of a product, so there are n product operations. Therefore, the total cost of the inner line is $2n$. Multiplying this by the number of times it is executed, we get $2n^3$, which is $O(n^3)$, the asymptotic cost of the standard algorithm for matrix multiplication.

3. Recursive algorithms. Here's a recursive algorithm, that given an n bit number in binary, $x = x_{n-1} \dots x_0$ and an $m < n$ bit number in binary y, computes $x \bmod y$. It uses the following sub-routines: $Add(x, y)$ adds two n bit binary numbers in $O(n)$ time, $AsLarge(x, y)$ returns true if and only if $x \geq y$ and also takes $O(n)$ time, and $Subtract(x, y)$ computes $x - y$ and is also $O(n)$ time. We assume $y \geq 2$ and $n \geq 1$.

x mod y

Mod $(x = x_{n-1} \dots x_1 x_0,\ y)$

1. IF $n = 1$, return x_0.
2. $v \leftarrow Mod(x_{n-1} \dots x_1,\ y)$.
3. $v \leftarrow Add(v, v)$

4. $v \leftarrow Add(v, x_0)$.
5. IF $AsLarge(v, y)$ THEN $v \leftarrow Subtract(v, y)$.
6. Return v

(a) **Correctness.** Prove by induction on n that $Mod(x, y)$ returns $x \mod y$.
Hints:
- Let $x = 2a + x_0$, where $a = x_{n-1} \dots x_1$.
- To show that two numbers are the same mod y, show that their difference is a multiple of y.

(b) **Recurrence.** Give a recurrence for $T(n)$, the time this algorithm takes on an input of size n.

(c) **Solving recurrences.** Solve this recurrence to give the Big O time of this algorithm.

Solution:

(a) First, we must prove correctness. Using the hints, we'll let $x = 2a + x_0$, and then we'll re-write a using the division theorem as $a = k_1 y + r_1$.

To prove a recursive algorithm correct, we need to use induction. The **base case** for this algorithm is $n = 1$ (invoking the algorithm on a 1-bit number). In this case, looking at line 1, we see that the algorithm just returns the single bit. This is correct, because a single-bit number can only be 0 or 1, and $0 \mod y = 0$ for any $y \geq 2$, and $1 \mod y = 1$ for any $y \geq 2$. We have this condition on y from the restrictions given on the input.

algorithm return vs expected x mod y

For the inductive case, we first assume that there is a $c > 1$ such that the Mod algorithm listed works for inputs of c bits (that is, for any x of c bits, $Mod(x, y)$ correctly returns $x \mod y$). This is the **inductive hypothesis**. Now, we must use this assumption to show that the algorithm is correct for inputs of $c + 1$ bits.

Note that $a = \lfloor \frac{x}{2} \rfloor$ is the c-bit number that Mod is called on during line 2, because we are dropping the least significant bit of x and "shifting" right. This is what allows us to write $x = 2a + x_0$.

First, let x be an arbitrary $c + 1$ bit number. Now, simulate the algorithm:

1. We know that the condition in line 1 will not be true, because $c + 1 > 1$, so the algorithm continues.
2. In line 2, we use the **IH** and the fact that a is c bits to assert that v is assigned the value $a \mod y$, which in our re-expression of a above is r_1. // v = a mod y
3. The value in v is doubled. //v=2(a mod y)
4. The value in v has x_0 added to it. //v=2(a mod y)+x0

After line 4, then, we have that:

$$v = 2r_1 + x_0$$

This algorithm returns v, possibly after subtracting y from it once. Since subtracting a multiple of y (such as $1 \times y$) will not change the value of $v \mod y$, all we need to do is prove that $v \mod y$ and $x \mod y$ are the same. Now we use the second hint: if we prove that $x - v$ is a multiple of y (that is, $\exists k'$ such that $x - v = k'y$) then we'll have that $v \mod y = x \mod y$ and the algorithm is correct after (possibly) adjusting the range of v to be between 0 and $y - 1$.

$(2r_1 + x_0)$

We'll show this by expanding the definitions of $x - y$. The key insight here is using the division theorem to rewrite v and a.

요거 어디서 나왔냐?

$2a + x_0$

$$
\begin{aligned}
x - v &= (2a + x_0) - (2r_1 + x_0) \\
&= (2(k_1 y + r_1) + x_0) - 2r_1 - x_0 \\
&= 2k_1 y + 2r_1 + x_0 - 2r_1 - x_0 \\
&= 2k_1 y \\
&= k' y
\end{aligned}
$$

a mod y

by IH.

Therefore, by the hint, $x \mod y = v \mod y$. To see why the hint is true, take arbitrary a and b and re-express them using the division theorem for y, and then subtract the re-expressions. The difference is a multiple of y iff a and b have the same remainder when divided by y.

The only thing we need to account for now is the size of v at the end of the algorithm: it could be as large as $2(y-1)$. The value that we want is a remainder after dividing by y, so it must be between 0 and $y - 1$. Because the v is maximized at $2(y-1)$ and because subtracting (or adding) multiples of y does not change a value modulo y, it suffices to subtract off just one y if v is too large.

To see why subtracting multiples of y does not change a value mod y, look again at the division theorem:

Express some a as $a = qy + r$, then subtract ly for some y, getting: $a - ly = (q - l)y + r$. The remainder has not changed, so the value mod y is the same.

(b) Now we write the following recurrence. We are told that $Add(x, y)$, $AsLarge(x, y)$, and $Subtract(x, y)$ all have $O(n)$ time, so from that we can say:

- *Add* has time $k_1 n$
- *AsLarge* has time $k_2 n$
- *Subtract* has time $k_3 n$

For some constants k_i. There is only one recursive call, on line 2. Note that *Add* is called twice. So the entire recurrence is:

$$
\begin{aligned}
T(n) &= T(n-1) + k_1 n + k_1 n + k_2 n + k_3 n \\
&= T(n-1) + (2k_1 n + k_2 n + k_3 n) \\
&= T(n-1) + kn
\end{aligned}
$$

For some $k = 2k_1 + k_2 + k_3$. Asymptotically, this is:

$$T(n) = T(n-1) + O(n)$$

With $T(1) = 1$ for the constant work involved in testing for a single bit and returning it.

(c) Now we solve the recurrence to obtain the runtime of the algorithm. Unwinding, look for a pattern:

$$\begin{aligned} T(n) &= T(n-1) + kn \\ &= T(n-2) + k(n-1) + kn \\ &= T(n-3) + k(n-2) + k(n-1) + kn \end{aligned}$$

Recognizing the pattern, we see that at each step of unrolling there is only one recursive term, and one new $k(n-i)$ term introduced into the sum. Eventually, when the term $T(n-(n-1))$ comes up, this will evaluate to 1 and the unrolling/recursion will stop. Therefore, we have:

$$\begin{aligned} T(n) &= \sum_{i=1}^{n} ki \\ &< k \sum_{i=1}^{n} i \\ &= k \times \frac{1}{2} \times n(n+1) \end{aligned}$$

is $O(n^2)$. Note that the $n-i$ terms were flipped to i by recognizing that this is the same set of numbers in the summation, just counted "backwards" instead of "forwards." This solves the recurrence and the runtime of the algorithms is therefore $O(n^2)$.

5. Recursive Counting

For problems (a)-(e), it is not required to solve the recurrence:

(a) In a round-robin tennis tournament with n players, every tennis player plays against every other player. Let $T(n)$ be the total number of tennis matches taking place among the n players. Find a recurrence for $T(n)$ and explain in words why $T(n)$ satisfies this recurrence.

Solution: We can think of a tournament with n people as a tournament with $n-1$ people, each playing against each other, plus one additional person who plays against each of the other $n-1$ people. So the recurrence is

$$T(n) = T(n-1) + (n-1), \text{ with } T(2) = 1$$

or equivalently,

$$T(n) = T(n-1) + (n-1), \text{ with } T(1) = 0.$$

(b) In a board game, players must place colored tiles in a single row. There are red 1×1 tiles, yellow 2×1 tiles, blue 2×1 tiles, purple 3×1 tiles, and green 3×1 tiles. Let $C(n)$ be the number of different $n \times 1$ colored rows that can be created using these tiles. Find a recurrence for $C(n)$ and explain in words why $C(n)$ satisfies this recurrence.

Solution: Consider the first colored tile in the row. If it is a red 1×1 tile, then the remainder of the row is just an $n-1 \times 1$ row of colored tiles, and the number of those is $C(n-1)$. If the first cell is a yellow 2×1 tile, then the remainder of the row is an $n-2 \times 1$ row of colored tiles, and the number of those is $C(n-2)$. We can use the same logic for each of the possible first tiles. Then the total number of ways to fill in the $n \times 1$ row is the number of ways starting with a red tile plus the number of ways starting with a yellow tile, etc. Therefore, the recurrence is

$$\begin{aligned}
C(n) &= C(n-1) + C(n-2) + C(n-2) + C(n-3) + C(n-3) \\
&= C(n-1) + 2C(n-2) + 2C(n-3), \\
&\text{with } C(0) = 1, C(1) = 1, C(2) = 3 \text{ or equivalently,} \\
C(n) &= C(n-1) + C(n-2) + C(n-2) + C(n-3) + C(n-3) \\
&= C(n-1) + 2C(n-2) + 2C(n-3), \\
&\text{with } C(1) = 1, C(2) = 3, C(3) = 7.
\end{aligned}$$

(c) Any binary string can be broken into contiguous chunks of the same character, called runs. For example, 001110100000 has:

- a run of 0s of length 2
- a run of 1s of length 3
- a run of 0s of length 1
- a run of 1s of length 1
- a run of 0s of length 5

Let $B(n)$ be the number of length n binary strings where each run of 1s has even length. Find a recurrence for $B(n)$ and explain in words why $B(n)$ satisfies this recurrence.

Solution: Every binary string either starts with 0 or 1. Suppose we have a binary string where each run of 1s has even length, and the string starts with 0. Then the remainder of the string is a length $n-1$ binary string where each run of 1s has even length, and there are $B(n-1)$ such strings. Now, suppose we have a binary string where each run of 1s has even length, and the string starts with 1. Then the second bit of the string must also be 1, otherwise the string would start with a run of length one, which is odd. After these first two 1s, the remainder of the string is a length $n-2$ binary string where each run of 1s has even length, and there are $B(n-2)$ such strings. Therefore, the recurrence is

$$B(n) = B(n-1) + B(n-2), \text{ with } B(0) = 1, B(1) = 1$$

or equivalently,

$$B(n) = B(n-1) + B(n-2), \text{ with } B(1) = 1, B(2) = 2.$$

(d) Let $S(n)$ be the number of subsets of $\{1, 2, \ldots, n\}$ having the following property: there are no three elements in the subset that are consecutive integers. Find a recurrence for $S(n)$ and explain in words why $S(n)$ satisfies this recurrence.

Solution: Consider all subsets of $\{1, 2, \ldots, n\}$ having the property that no three elements in the subset are consecutive integers. Each such subset falls in one of two cases, based on whether the subset contains the largest possible integer, n.
Case 1: The subset does not contain n. In this case, we can think of the subset as being a subset of $\{1, 2, \ldots, n-1\}$ having the property that no three elements in the subset are consecutive integers. Therefore, the number of subsets that fall into this case is $S(n-1)$.

Case 2: The subset contains n. Then this subset falls into one of the following cases, based on whether it also contains $n-1$:
Case 2a: The subset contains n and $n-1$. In this case, we know $n-2$ must not be an element of the subset, otherwise it would contain three consecutive integers. Then, we can think of the subset as a subset of $\{1, 2, \ldots, n-3\}$ having the property that no three elements in the subset are consecutive integers, together with n and $n-1$. Therefore, the number of subsets that fall into this case is $S(n-3)$.
Case 2b: The subset contains n but not $n-1$. In this case, we can think of the subset as being a subset of $\{1, 2, \ldots, n-2\}$ having the property that no three elements in the subset are consecutive integers, together with n. Therefore, the number of subsets that fall into this case is $S(n-2)$.
Then the recurrence is

$$S(n) = S(n-1) + S(n-2) + S(n-3), \text{ with } S(0) = 1, S(1) = 2, S(2) = 4$$

or equivalently,

$$S(n) = S(n-1) + S(n-2) + S(n-3), \text{ with } S(1) = 2, S(2) = 4, S(3) = 7.$$

4. **Counting.** (a) How many 5-card hands can be formed from an ordinary deck of 52 cards if exactly two suits are present in the hand?

(b) In any bit string, the longest consecutive run length is the maximum number of consecutive 1's or consecutive 0's in the string. For example, in the string 1101000111, the longest consecutive run length is 3. How many bit strings of length 10 have a longest consecutive run length of 6?

(c) A software company assigns its summer interns to one of three divisions: design, implementation, and testing. In how many ways can a group of ten interns be assigned to these divisions if each division needs at least one intern?

(d) How many numbers in the interval [1,10000] are divisible by 7, 9, or 11?

(e) A California license plate follows the format $DLLLDDD$, where D represents a digit and L represents an upper case letter. For example, a valid California license plate is $5JBC434$ (Quang's plate). How many California license plates are possible?

Under a fixed-length encoding scheme, how many bits of memory does the California DMV need to allocate in its database to store each license plate number?

Solution:

(a) There are two disjoint possibilities to have exactly two suits:

- 3 cards of one kind + 2 cards of another: The steps involved are:
  (a) Select a kind (k_1) for 3 cards - $\binom{4}{1}$
  (b) Select a kind (k_2) for 2 cards - $\binom{3}{1}$
  (c) Select 3 cards from kind k_1 - $\binom{13}{3}$
  (d) Select 2 cards from kind k_2 - $\binom{13}{2}$
  Total number of ways is $\binom{4}{1} \cdot \binom{3}{1} \cdot \binom{13}{3} \cdot \binom{13}{2}$
- 4 cards of one kind + 1 card of another kind: Using similar counting as previous case, $\binom{4}{1} \cdot \binom{3}{1} \cdot \binom{13}{4} \cdot \binom{13}{1}$

Total number of ways is: $\binom{4}{1}\cdot\binom{3}{1}\cdot\binom{13}{3}\cdot\binom{13}{2}+\binom{4}{1}\cdot\binom{3}{1}\cdot\binom{13}{4}\cdot\binom{13}{1}$

(b) Let $c_1, c_2, c_3, \ldots, c_{10}$ be the 10 spots for 10 digits. We need to fill these spots with 0, 1 such that there are exactly 6 consecutive spots with same number. There are two cases:

- The six consecutive spots are at the beginning or the end. That is, starting location of six consecutive spots is either c_1 or c_5. All the consecutive spots need to take same value and the adjacent spot should take the opposite value. This can be done in two ways. The remaining 3 spots can take any of the two values each (or in 2^3 ways). So, total possibilities is $2 \cdot 2 \cdot 2^3 = 32$
- The six consecutive spots are in the middle. That is, starting location of size consecutive spots is one of c_2, c_3, c_4. All the consecutive spots need to take same value and the two adjacent spots should take the opposite values. This can be done in two ways. The remaining 2 spots can take any of the two values each (or in 2^2 ways). So, total possibilities is $3 \cdot 2 \cdot 2^2 = 24$

So, total number of ways is 56.

(c) We'll use the principle of inclusion-exclusion for this problem.

Each person can be assigned to one of the three divisions, so for 10 people the total number of ways of assigning interns to divisions is 3^{10}.

Now let's count the number of ways that all of them go to just two particular divisions, say design and implementation. We have to exclude this case from our count because testing will be empty. There are two possible assignments for each person, so the number of possibilities is 2^{10}. Similary, we have to account for other two cases where everyone goes to implementation and testing or everyone goes to design and testing. In total, there are $3 \cdot 2^{10}$ ways that the interns can be assigned to only two divisions.

Finally, we must compute the number of ways in which interns are assigned to only one division. There are 3 ways to select which single division will have all interns, and this is the only choice we can make.

Applying the principle of inclusion-exclusion, the number of ways the interns can be assigned to the divisions so that each division has at least 1 intern is given by the total number of possible assignment, minus the number of ways to assign interns to only 2 divisions, plus the number of ways to assign interns to 3 divisions, which is $3^{10} - 3 \cdot 2^{10} + 3$.

(d) Use inclusion-exclusion with three sets:
$A = \{$ numbers in the interval $[1, 10000]$ which are divisible by $7\}$,
$B = \{$ numbers in the interval $[1, 10000]$ which are divisible by $9\}$, and
$C = \{$ numbers in the interval $[1, 10000]$ which are divisible by $11\}$

We have
$|A| = \left\lfloor \dfrac{10000}{7} \right\rfloor = 1428,$
$|B| = \left\lfloor \dfrac{10000}{9} \right\rfloor = 1111,$ and
$|C| = \left\lfloor \dfrac{10000}{11} \right\rfloor = 909.$

To compute the intersection of two of these sets, notice that since $7, 9$, and 11 have no common factors (they are relatively prime), a number that is divisible by two of $7, 9$, and 11 must be divisible by their product. For example, a number being divisible by 7 and 11 just means it is divisible by 77. So,

$|A \cap B| = \left\lfloor \dfrac{10000}{63} \right\rfloor = 158,$
$|A \cap C| = \left\lfloor \dfrac{10000}{77} \right\rfloor = 129,$ and
$|B \cap C| = \left\lfloor \dfrac{10000}{99} \right\rfloor = 101.$

Similarly, if a number is divisible by $7, 9,$ and 11, it must be divisible by their product 693, and vice versa. Therefore,
$|A \cap B \cap C| = \left\lfloor \dfrac{10000}{693} \right\rfloor = 14.$

Putting this all together, the inclusion-exclusion formula gives:

$$\begin{aligned} |A \cup B \cup C| &= |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C| \\ &= 1428 + 1111 + 909 - 158 - 129 - 101 + 14 \\ &= 3074. \end{aligned}$$

(e) There are 26 upper case letters and 10 digits. Using the Product Rule and the number of options in each position gives $10 * 26 * 26 * 26 * 10 * 10 * 10 = 175760000$. The number of bits needed to store each object when there are n objects is $\lceil \log_2 n \rceil$. So the number of bits of memory needed to store each license plate is $\lceil \log_2 175760000 \rceil = \lceil 27.389 \rceil = 28$.

5. Probability. (a) I have 10 shirts, 6 pairs of pants, and 3 jackets. Every day I dress at random, picking one of each category. What is the probability that today I am wearing at least one garment I was wearing yesterday?

(b) A permutation of size n is a rearrangement of the numbers $\{1, 2, \ldots, n\}$ in any order. A *rise* in a permutation occurs when a larger number immediately follows a smaller one. For example, if $n = 5$, the permutation 1 3 2 4 5 has three rises. What is the expected number of rises in a permutation of size n?

(c) There are n teams in a sports league. Over the course of a season, each team plays every other team exactly once. If the outcome of each game is a fair random coin flip, and there are no ties, what is the probability that some team wins all of its games?

(d) Suppose there are 10 people of distinct heights standing in line for free food, arranged in a random order. Use linearity of expectation to calculate the expected number of people who can see all the way to the front of the line.

(e) Assume that every time you go on a job interview, your chance of getting a job offer is 25%. How many job interviews must you go on so that the probability of your getting a job offer is greater than 95%?

(f) Suppose that you roll two dice and don't get to look at the outcome. Your friend looks at the outcome and tells you honestly that at least one of the dice came up 6. What is the probability that the sum of your two dice is 8?

(g) Suppose there are n people assigned to m different tasks. Assume that each person is randomly assigned a task and that for each person, all tasks are equally likely. Use linearity of expectation to find the expected number of people working on a task alone.

(h) Tesla is planning to introduce a new self-driving car. The company commissions a marketing report for each new car that predicts either the success or the failure of the car. Of the new cars introduced by the company so far, 65% have been successes. Furthermore, 70% of their successful cars were predicted to be successes, while 40% of failed cars were predicted to be successes. Find the probability that this new self-driving car will be successful if its success has been predicted.

Solution:

(a) Let us instead compute the complement, since this is a question about whether there exists a match between today's clothing and yesterday's.

Let A be the event that I am wearing all different clothes from yesterday.

We can see that $P(A) = (9/10) * (5/6) * (2/3)$. The reason for this is that today I can wear any of the other garments that I didn't wear yesterday.

Thus the probability of wearing at least one of the same garments as yesterday is $1 - P(A) = 1 - (9/10) * (5/6) * (2/3)$.

each: i

(b) We want $E[X]$ where X is the number of rises in the permutation. However, the number of rises in the permutation is the total number of rises that start at each position. So for $i = 1$ to $n-1$, define X_i to be 1 if there is a rise starting at position i and 0 otherwise. Thus, using linearity of expectation,

$$E[X] = E[X_1 + X_2 + \cdots + X_{n-1}] = E[X_1] + E[X_2] + \cdots + E[X_{n-1}].$$

If we think of a permutation as a random rearrangement of the numbers from 1 to n, no position in a permutation is more likely to have a rise than any other. This means each X_i has the same distribution and the same expected value. With probability $1/2$ there is a rise at position i and with probability $1/2$ there is a fall, so $E[X_i] = 1*1/2+0*1/2 = 1/2$. Thus, $E[X] = (n-1) * 1/2$.

(c) As there n teams in the league, each team plays $n-1$ games, so for a team to win all of its games it has to win all $n-1$ games. Every game to be won has a probability of $1/2$.

Define i events, E_i is the event that team i is undefeated. The event that some team is undefeated is $E_1 + E_2 + \cdots + E_n$, so we want $P(E_1 + E_2 + \cdots + E_n)$. But the events E_i are mutually exclusive since we can't have two undefeated teams, which means we can add their probabilities together and instead compute $P(E_1) + P(E_2) + \cdots + P(E_n)$.

The probability that team i wins all of its games is $P(E_i) = (1/2)^{n-1}$. As there are n teams the total probability that some team is undefeated is $n * (1/2)^{n-1}$.

(d) Let X be the random variable that represents the number of people who can see to the front. Suppose each person is numbered $1, 2, \ldots, 10$. For each i from 1 to 10, let X_i be the random variable defined to be $X_i = 1$ if person i can see to the front and $X_i = 0$ otherwise.

For each i from 1 to 10, $E(X_i) = 1/i$ because for person i to see to the front, he must be taller than all the people before him. So the tallest of persons 1 through i must wind up in position i. The probability that the tallest of the first i people winds up in position i is $1/i$, since there are i positions where the tallest person can go, and only one of them is position i. Therefore $P(X_i = 1) = 1/i$ and $P(X_i = 0) = (1 - 1/i)$ for each person i. Therefore the expected value is $E(X_i) = (1)(1/i) + (0)(1 - 1/i) = 1/i$ for each person i.

It follows that $X = \sum_{i=1}^{10} X_i$ and that $E(X) = E\left(\sum_{i=1}^{10} X_i\right) = \sum_{i=1}^{10} E(X_i)$.

Therefore, $E(X) = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \frac{1}{10} = \frac{7381}{2520}$

(e) Suppose that A is the event that you get the job offer. Then A^c is the event that you do not get the job offer. $P(A) = 1/4$ and $P(A^c) = 1 - 1/4 = 3/4$. The probability that after n attempts you did not get offered the job is $P(A^c)^n = 3^n/4^n$. So the probability that after n attempts, you did get offered the job is $1 - P(A^c)^n = 1 - 3^n/4^n$. So the solution of this problem is to solve for n such that $1 - 3^n/4^n > 95/100$.

$$1 - 3^n/4^n > 95/100 \implies 4^n/3^n > 100/5 \implies n > \log_{4/3}(100/5) \approx 10.8$$

So, it would take 11 attempts or more to ensure that you have a 95% chance of getting a job offer.

(f) Let A be the event that at least one of your rolls is a 6 and let B be the event that the sum of your two dice is 8. The problem is to compute the probability that B occurs given that A has occurred, which is $P(B|A) = \dfrac{P(B \cap A)}{P(A)}$.
A^c is the event that neither of your rolls are 6 and $P(A^c) = (5/6)^2$. Therefore $P(A) = 1 - (5/6)^2 = 11/36$. $B \cap A$ is the event that at least one of your dice is 6 and the sum is 8. It is clear that $B \cap A = \{(6,2),(2,6)\}$ and the sample space has 36 elements therefore $P(B \cap A) = 2/36$.
So the answer is $P(B|A) = \dfrac{P(B \cap A)}{P(A)} = \dfrac{2/36}{11/36} = 2/11$.

(g) Let X be the random variable that represents the number of people that are working on a task alone. The solution is to compute the expected value $E(X)$. Suppose the people are numbered $1, 2, \ldots, n$. Let X_i be the random variable such that $X_i = 1$ if person i is alone and $X_i = 0$ otherwise. It follows that $X = \sum_{i=1}^n X_i$. Therefore $E(X) = E\left(\sum_{i=1}^n X_i\right) = \sum_{i=1}^n E(X_i)$.
What is the probability that person i is alone? Person i is assigned a certain task and for each other person there is a $\frac{m-1}{m}$ chance that they are doing a different task than person i. So the chance that all $n-1$ other people are doing a different task than person i is $P(X_i = 1) = \left(\frac{m-1}{m}\right)^{n-1}$. So we have that the expected value of each X_i is $E(X_i) = \left(\frac{m-1}{m}\right)^{n-1}$.

$$E(X) = \sum_{i=1}^n E(X_i) = \sum_{i=1}^n \left(\frac{m-1}{m}\right)^{n-1} = n\left(\frac{m-1}{m}\right)^{n-1}.$$

(h) event that the car's success has been predicted and let F be the event that the car will be a success. Then we want to compute $P(F|E)$. Using Bayes' theorem, we have that

$$P(F|E) = \frac{P(E|F)P(F)}{P(E|F)P(F) + P(E|\bar{F})P(\bar{F})}.$$

According to the problem, $P(F) = 0.65$, $P(\bar{F}) = 0.35$, $P(E|F) = 0.70$, $P(E|\bar{F}) = 0.40$. So

$$P(F|E) = \frac{(0.7)(0.65)}{(0.7)(0.65) + (0.4)(0.35)} \approx 0.76.$$

(i) We can compute the variance with the following formula: $V(X) = E\left(X^2\right) - \left(E(X)^2\right)$. There are only two outcomes for flipping a coin once: namely heads and tails. So $X_1(H) = 1$ and $X_1(T) = 0$.

$$E(X_1) = X_1(H) * P(H) + X_1(T) * P(T) = p$$

$$E\left(X_1^2\right) = \left(X_1(H)^2\right) * P(H) + \left(X_1(T)^2\right) * P(T) = p$$

So the variance is:

$$V(X_1) = E\left(X_1^2\right) - E(X_1)^2 = p - p^2 = p(1-p).$$

Since each coin toss is independent, and $X_n = X_1 + X_1 + \cdots + X_1 = nX_1$, then

$$V(X_n) = nV(X_1) = np(1-p).$$

6. **Representing problems as graphs.** You have a system of n variables representing real numbers, $X_1, \ldots, X_n$. You are given a list of inequalities of the form $X_i < X_j$ for some pairs i and j. You want to know whether you can deduce with certainty from the given information that $X_1 < X_n$.

   For example, say n is 4. A possible input is the list of inequalities $X_1 < X_2, X_1 < X_3, X_4 < X_3$ and $X_3 < X_2$. Does it follow that $X_1 < X_4$?

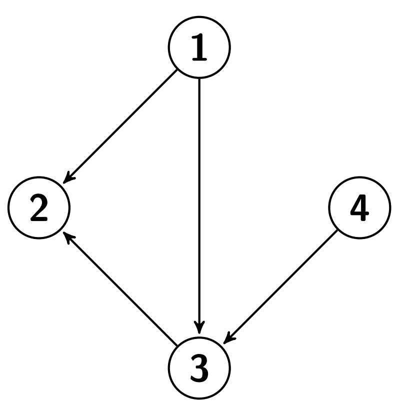
   (a) Give a description of a directed graph that would help solve this problem. Be sure to define both the vertices and edges in terms of the variables and known inequalities.

   (b) Draw the graph you described for the example above. Does $X_1 < X_4$ follow? Why or why not?

   (c) Say which algorithm from lecture we could use on such a graph to determine whether $X_1 < X_n$ follows from the known inequalities.

   Solution:

   (a) There will be n vertices labeled $1, 2, \ldots, n$. Connect vertex i to vertex j by a directed edge from i to j if and only if $X_i < X_j$. This helps solve the problem because we can deduce that $X_i < X_j$ if there exists a path from i to j in the graph.



   (b) It doesn't follow that $X_1 < X_4$ because there is no path from 1 to 4 in the graph.

   (c) We could use the Graph Search algorithm to decide whether or not n is reachable from 1. If it is, then $X_1 < X_n$. Otherwise we can't be sure of the relationship between X_1 and X_n.

7. **Representing problems as graphs.** We say a matrix has dimensions $m \times n$ if it has m rows and n columns. If matrix A has dimensions $x \times y$ and matrix B has dimensions $z \times w$, then the product AB exists if and only if $y = z$. In the case where the product exists, AB will have dimensions $x \times w$. In this problem, we are given a list of matrices and their dimensions, and we want to determine if there is an order in which we can multiply all the matrices together, using each matrix exactly once. The list of matrices and their dimensions is as follows:

$$\begin{array}{l} \text{A is } 4 \times 5, \\ \text{B is } 5 \times 7, \\ \text{C is } 5 \times 4, \\ \text{D is } 4 \times 7, \\ \text{E is } 7 \times 5. \end{array}$$

(a) Draw a graph that represents this situation in such a way that finding an order we seek corresponds to finding a Hamilton Tour of your graph. Describe what the vertices of your graph represent, and when two vertices are connected with an edge.

(b) Draw a different graph that represents this situation in such a way that finding an order we seek corresponds to to finding an Euler Tour of your graph. Describe what the vertices of your graph represent, and when two vertices are connected with an edge.

(c) Give an order in which we can multiply these matrices, or say that no such order exists.

Solution:

(a) One way is to use each matrix as a vertex, and connect matrix X to matrix Y if the matrix product XY exists. In this setting, we want a Hamilton tour that uses each vertex (each matrix) exactly once.

(b) Another way is to use each dimension that shows up in the list as a vertex, in this case $4, 5, 7$. Connect vertex m to n if there is an $m \times n$ matrix in the list. In this setting, an order that we want corresponds to an Euler Tour because we must use each edge (each matrix) exactly once.

(c) DECAB, ABECD, ACDEB are all possible answers.

8. **Eulerian and Hamiltonian path/circuit.** Han and Chewie are on the run from the Imperial Army. Starting from the Rebel's base on Yavin 4, in order to shake off Darth Vader's pursuit, they have to make a series of hyperspace jumps to seven different planets then return to the original base. However, the Millennium Falcon cannot make any jump longer than 12 *parsec* (Note: *parsec* is a unit of distance, not time). They looked up their star map and made the following table of the distances between planets. All the entries below are measured in *parsec*.

	Yavin 4	Coruscant	Tatooine	Hoth	Endor	Naboo	Kessel	Dagobah
Yavin 4	0	10	15	15	18	14	16	12
Coruscant	10	0	12	13	16	12	14	10
Tatooine	15	12	0	10	12	9	10	12
Hoth	15	13	10	0	12	12	13	15
Endor	18	16	12	12	0	12	11	16
Naboo	14	12	9	12	12	0	9	12
Kessel	16	14	10	13	11	9	0	14
Dagobah	12	10	12	12	16	12	14	0

(a) Draw a graph that will help Han and Chewie plan their escape. Specify whether your graph is directed or undirected. Carefully describe what the vertices of your graph represent, and when two vertices are connected with an edge.

(b) Say which graph theory problem they are trying to solve on this graph.

(c) Find a feasible escape route for Han and Chewie that meets the constraints, or explain why no such route exists.

Solution:

(a) This is an undirected graph. Here each vertex represents a planet. We put an edge between two vertices if an only if the distance between the two corresponding planets is less than or equal to 12 *parsec*.

(b) Han and Chewie are trying the solve the **Hamiltonian Circuit/Cycle problem**. They wish to find a path which visits every vertex in a graph exactly once and is also a cycle (start and end at the same vertex - Yavin 4).

(c) There are many answers. Here is one such path: Yavin 4 $\rightarrow$ Coruscant $\rightarrow$ Tatooine $\rightarrow$ Kessel $\rightarrow$ Endor $\rightarrow$ Hoth $\rightarrow$ Naboo $\rightarrow$ Dagobah $\rightarrow$ Yavin 4.

9. DAGs A daily flight schedule is a list of all the flights taking place that day. In a daily flight schedule, each flight F_i has an origin city OC_i, a destination city DC_i, a departure time d_i and an arrival time $a_i > d_i$. This is an example of a daily flight schedule for June 08, 2019 (our exam date), listing flights as $F_i = (OC_i, DC_i, d_i, a_i)$:

$$
\begin{aligned}
F_1 &= (\text{Portland, Los Angeles, 7:00am, 9:00am}) \\
F_2 &= (\text{Portland, Seattle, 8:00am, 9:00am}) \\
F_3 &= (\text{Los Angeles, San Francisco, 8:00am, 9:30am}) \\
F_4 &= (\text{Seattle, Los Angeles, 9:30am, 11:30am}) \\
F_5 &= (\text{Los Angeles, San Francisco, 12:00pm, 1:00pm}) \\
F_6 &= (\text{San Francisco, Portland, 1:30pm, 3:00pm})
\end{aligned}
$$

(a) Given any daily flight schedule, describe how to construct a DAG so that paths in the DAG represent possible sequences of connecting flights a person could take. What are the vertices, and when are two vertices connected with an edge?

(b) Why is your graph always a DAG?

(c) What problem would you need to solve in your DAG to help you determine the maximum number of flights a person could take on a given day?

(d) Draw the DAG you described for the given example of June 08, 2019 and give the maximum number of flights a person could take on that day.

Solution.

(a) The vertices are the flights in the daily flight schedule. Draw a directed edge from flight F_i to flight F_j if

- $DC_i = OC_j$ (flight F_j leaves from where flight F_i lands), and

- $a_i \le d_j$ (flight F_j leaves after flight F_i lands).

(b) Our graph is always a DAG because it's clearly a directed graph, and a cycle would be impossible since that would require you to take the same flight more than once on a given day. This is impossible because once you have taken a flight, some time has elapsed, and you cannot go back in time to take the same flight again.

More formally, we can say that if there is cycle involving F_i and F_j then there must be a directed path from F_i to F_j and from F_j to F_i. Then according to our definition of edges, a directed path from F_i to F_j means $a_i \le d_j$ and $a_j \le d_i$. Also, by definition, $a_i > d_i$ and $a_j > d_j$. But then putting these together we have $a_i \le d_j < a_j \le d_i$ which is a contradiction to the fact that $a_i > d_i$.

(c) Since paths in the DAG represent possible sequences of connecting flights, the maximum number of flights a person could take is determined by the longest path in the graph.

Using the graph shown above, we see that the longest path includes four flights (F_2, F_4, F_5, F_6), so the maximum number of flights a person could take on that day is four.

10. Trees. A binomial tree is a special kind of rooted tree used for various data structures in computer science. A degree d binomial tree can be defined recursively as follows. A degree 0 binomial tree is a single vertex with no edges. A degree d binomial tree has a root vertex with out-degree d. The first (that is, leftmost) subtree is a degree $d-1$ binomial tree. The second (that is, second to left) subtree is a degree $d-2$ binomial tree. Continue on in this way so that the last (rightmost) subtree is a degree 0 binomial tree.

(a) What is the height of a degree d binomial tree? Prove your result by induction on d.

(b) Write a recurrence for the number of nodes $N(d)$ in a binomial tree of degree d.

(c) Use the guess-and-check method to guess a formula for $N(d)$. Prove that your formula holds by induction on d.

Solution:

(a) A binomial tree of degree d has a height of d. We prove this statement by strong induction on d.

When $d = 0$ the tree is just one node, so it has a height of 0.

Now let $d \ge 0$. Assume as the inductive hypothesis that for all $0 \le k \le d$, a binomial tree of degree k has a height of k.

Consider a binomial tree of degree $d+1$. Our root has one child each of degree d,d-1, . . . 0. By our induction hypothesis, for each value of k, $0 \le k \le d$, the subtree rooted at the child of degree k has a height k. Therefore the maximum height of one of these subtrees is d (choosing the largest value of k). Since our binomial tree of degree $d+1$ has a subtree of height d, then it must have a height of $d+1$.

(b)

$$N(d) = N(d-1) + N(d-2) + \cdots + N(0) + 1, N(0) = 1$$

This comes from counting the number of nodes in each subtree, from left to right, plus one additional root node.

(c) We guess that $N(d) = 2^d$, and we prove the result by strong induction on d.

When $d = 0$ the tree is just one node so $N(0) = 1$, which satisfies the formula since $2^d = 2^0 = 1$.

Now let $d \geq 0$. Assume as the inductive hypothesis that for all $0 \leq k \leq d$, $N(k) = 2^k$. Then

$$\begin{aligned} N(d+1) &= N(d) + N(d-1) + N(d-2) + \cdots + N(1) + N(0) + 1 \\ &= 2^d + 2^{d-1} + 2^{d-2} + \cdots + 2^1 + 2^0 + 1 \quad \text{(by IH)} \\ &= 2^{k+1} \quad \text{(by a formula for the sum of powers of two, or by geometric series)} \end{aligned}$$

11. Stars and Bars item Let n, k be positive integers. Recall the *Stars and Bars Problem* that the integer equations $a_1 + a_2 + \ldots + a_k = n$, where $a_i \geq 0$, has $\binom{n+k-1}{k-1}$ solutions.

Find the number of integer solutions of the equation $x_1 + x_2 + x_3 = 50$ such that $0 \leq x_i \leq 19$, for $i = 1, 2, 3$. You may leave your answer in terms of the binomial coefficients.

(*Hint: Let A_i be the set of all solutions in which $x_i \geq 20$.*)

Solution. Without the upper-bound restriction, the number of non-negative integer solutions to the equation $x_1 + x_2 + x_3 = 50$ is given by $\binom{52}{2}$.

Let A be the set of solutions such that at least one of the x_i's is greater or equal to 20. Then these are the "bad" solutions. By inclusion exclusion:

$$|A| = |A_1| + |A_2| + |A_3| - |A_1 \cap A_2| - |A_2 \cap A_3| - |A_3 \cap A_1| + |A_1 \cap A_2 \cap A_3|$$

Also notice that $|A_1| = |A_2| = |A_3|$ and $|A_1 \cap A_2| = |A_1 \cap A_3| = |A_2 \cap A_3|$.

To find $|A_1|$, define $x_1' = x_1 - 20$ then the equation $x_1 + x_2 + x_3 = 50$ now becomes $x_1' + x_2 + x_3 = 30$ where $x_1', x_2, x_3 \geq 0$. By Stars and Bars, there are $\binom{32}{2}$ solutions (x_1', x_2, x_3). Since there is a one-to-one correspondent between the triples (x_1', x_2, x_3) and (x_1, x_2, x_3), we know that $|A_1| = |A_2| = |A_3| = \binom{32}{2}$.

To find $|A_1 \cap A_2|$, define $x_1' = x_1 - 20, x_2' = x_2 - 20$ then the equation $x_1 + x_2 + x_3 = 50$ now becomes $x_1' + x_2' + x_3 = 10$ where $x_1', x_2', x_3 \geq 0$. By Stars and Bars, there are $\binom{12}{2}$ solutions (x_1', x_2', x_3). Similar to the previous section, we know that $|A_1 \cap A_2| = |A_1 \cap A_3| = |A_2 \cap A_3| = \binom{12}{2}$.

Finally, $|A_1 \cap A_2 \cap A_3| = 0$ since it is impossible to have three non-negative integers whose sum is 50 while each integer is at least 20.

Hence, the number of bad solutions is $|A| = 3 * \binom{32}{2} - 3 * \binom{12}{2}$.

Thus, the final answer is $\binom{52}{2} - 3 * \binom{32}{2} + 3 * \binom{12}{2}$.