CSE 21 – Winter 2018 – Final Exam Solution

1. Order Notations (10 points)

(a) (6 points) Place the six functions below in the appropriate blanks to create a list of functions such that each function is big-O of the next function. No justification needed.

$n^{1.2}$ $\quad (\log n)^2 \quad (n-2)! \quad 3^{2n} \quad n\sqrt{n} \quad \log(\log n)$

$$1,\ \underline{\log(\log n)}\ ,\ \underline{(\log n)^2},\ n,\ \underline{n^{1.2}}\ ,\ \underline{n\sqrt{n}},\ n^2,\ \underline{3^{2n}}\ ,\ \underline{(n-2)!},\ n^n$$

(b) (4 points) Circle true or false for each statement. No justification is needed.

(a) $(2n^2 + 2n + 3)^3 \in \Theta((3n^3 + n^2 + 2n + 2)^2)$

True False

(b) If $f(n) \in \Theta(g(n))$, then $2^{f(n)} \in \Theta(2^{g(n)})$.

True **False**

(c) If $f(n) \in \Omega(n^3)$, then $f(n) \in \Omega(n^4)$.

True **False**

(d) $n^3 \log_2 n \in O(n \log_2 n^3)$

True **False**

2. Analyzing Algorithms (10 points)

(a) (4 points) Consider the following algorithm:

procedure *Count*(n: a positive integer)

```
x := 0
m := n + 10
for i := 1 to 5n
    for j := n/5 to n
        for k := n to m
            x := x - 1
return x
```

What is the output of $Count(10)$? Answer: -4950

(b) (b.1.) (3 points)

procedure Loops1(n, a positive integer)

```
for i = 1 to n
    for j = i + 1 to n
        print (i, j)
```

The exact number of ordered pairs printed, in terms of n:

$(n-1)+(n-2)+\cdots+2+1 = n(n-1)/2$

The runtime in Θ notation of the procedure: $\Theta(n^2)$

(b.2.) (3 points)

procedure Loops2(n, a power of 2)

```
for i = 1 to n
    j = n
    while j > 1
        print (i, j)
        j = j/2
```

The exact number of ordered pairs printed, in terms of n: $n\log_2(n)$

The runtime in Θ notation of the procedure: $\Theta(n\log(n))$

3. Recursive Algorithm & Solving Recurrence (10 points)

(a) (5 points) Let n be a positive integer. We are given as input two words of the same length n and we want to create a word of length $2n$ that comes from alternating the letters of each word, starting with the first word. For example, on an input of $(hello, world)$ the algorithm should produce the output *hweolrllod*. Here is a recursive algorithm to solve this problem.

procedure *Interleave*($q_1q_2 \ldots q_n$: word of length n, $r_1r_2 \ldots r_n$: word of length n)

1. **if** $n = 1$ **then**
2. **return** q_1r_1
3. *output* := $Interleave(q_1q_2 \ldots q_{n-1}, r_1r_2 \ldots r_{n-1})$
4. Append q_nr_n to the end of *output*.
5. **return** *output*

Prove by induction on n that *Interleave* is correct.

<u>Rubric:</u>

- **2 points** for the base case $n = 1$.
- **1 point** for stating the induction hypothesis: "Suppose the algorithm is true on an input of size k (or $k-1$), we want to show it is also true on an input of size $k+1$ (or k)."
- **2 points** for proving the inductive step.

(b) A permutation of length n is a rearrangement of the numbers $\{1, 2, \ldots, n\}$. A permutation is called *unimodal* if the numbers reading from left to right first increase and then decrease.

For example, the permutation 1 2 3 5 7 6 4 is a unimodal permutation of length 7, and 2 5 4 3 1 is a unimodal permutation of length 5.

We also consider increasing permutations of the form 1 2 3 $\cdots$ n and decreasing permutations of the form n $\cdots$ 3 2 1 to be unimodal permutations. For example, 1 2 3 4 and 4 3 2 1 are unimodal permutations of length 4.

(b.1.) (2 points) For $n \geq 1$, let $S(n)$ be the number of unimodal permutations of length n. Write a recurrence relation that $S(n)$ satisfies. Include all the base case(s).

Answer: $S(n) = 2S(n-1)$ with $S(1) = 1$.

(b.2.) (3 points) Solve the recurrence in part (a) to find a closed-form formula for $S(n)$.

Answer: $S(n) = 2^{n-1}$

Students can use either unraveling or guess-and-check. If they use the latter, then they must prove their formula by induction

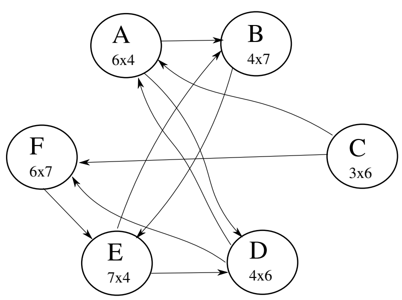
4. Representing Problems as Graph (10 points)

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. For example, here is a possible list of matrices and their dimensions:

A is 6×4
B is 4×7
C is 3×6
D is 4×6
E is 7×4
F is 6×7

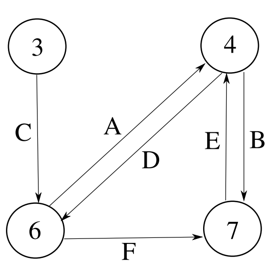
(a) (4 points) Given any list of matrices and dimensions, describe how to draw a graph so that finding an order in which we can multiply the matrices corresponds to finding a **Hamiltonian path** of your graph. Carefully describe what the vertices of your graph represent, identify whether the graph is directed or undirected, and specify when two vertices are connected with an edge.

Answer: Let each matrix be a vertex. Draw a directed edge from matrix M to matrix N if the number of columns of matrix M equals the number of rows of matrix N, or in other words, if the matrix product MN is defined.



(b) (4 points) Given any list of matrices and dimensions, describe how to draw a graph so that finding an order in which we can multiply the matrices corresponds to finding an **Eulerian path** of your graph. Carefully describe what the vertices of your graph represent, identify whether the graph is directed or undirected, and specify when two vertices are connected with an edge.

Answer: Make a vertex for each number that appears as a dimension of a matrix in the list. Draw a directed edge from vertex i to vertex j if there is an $i \times j$ matrix in the list.



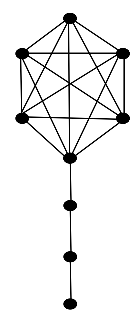
(c) (2 points) For the given example list of matrices above, give one order in which we can multiply those matrices, or say that no such order exists.

Answer: There are many possible orders in which we could multiply the example list of matrices, one of which is $CABEDF$

5. Graphs, Tours, DAGs, and Trees (10 points)

(a) For n an integer with $n \geq 3$, let *Lollipop*(n) be the graph that is made by attaching a chain of three additional vertices to a complete graph with n vertices.

A complete graph with n vertices is a graph in which every vertex is adjacent to every other vertex besides itself. As an example, the graph *Lollipop*(6)



The next two questions are about *Lollipop*(n) for general $n \geq 3$.

(1.1.) (2 points) For which values of $n \geq 3$ does the graph *Lollipop*(n) have an **Eulerian circuit**? Justify your answer.

Answer: We can **never** have a Eulerian circuit since there is always a vertex with odd degree - at the bottom.

(2.2.) (3 points) For which values of $n \geq 3$ does the graph *Lollipop*(n) have an **Eulerian path**? Justify your answer.

Answer: We want $n \geq 3$ to be an **odd** number (i.e. $3, 5, 7, 9, \ldots$) so now we have exactly two vertices with odd degree.

(b) (5 points) There is no partial credit for the following problems, so you do not need to show any work. Write your answer in terms of n or d.

(b.1.) What is the **number of edges** of a binary tree with n vertices?

Answer: $n - 1$

(b.2.) What is the **minimum height** of a binary tree with n vertices?

Answer: $\lfloor \log_2(n) \rfloor$

(b.3.) What is the **maximum height** of a binary tree with n vertices?

Answer: $n - 1$

(b.4.) What is the **minimum number of vertices** in a binary tree of height d?

Answer: $d + 1$

(b.5.) What is the **maximum number of vertices** in a binary tree of height d?

Answer: $2^{d+1} - 1$

6. Inclusion - Exclusion (10 points)

A valid password must be exactly **7** characters in length. They can contain upper case letters $\{A, B, \ldots, Z\}$, lower case letters $\{a, b, \ldots, z\}$, and numbers $\{0, 1, \ldots, 9\}$. Each password must contain **at least one upper case letter, at least one lower case letter, and at least one number**. How many passwords are possible in this system?

Note: You may leave your answers as unsimplified expressions with factorials, exponents, binomial coefficients, etc. There are 26 letters in the English alphabet.

Solution: Let

- $\mathcal{S}$ be the set of all passwords of length 7, both valid and invalid. Then $|\mathcal{S}| = 62^7$
- A be the set of length 7 passwords that do not contain an upper case letter. Then $|A| = 36^7$
- B be the set of length 7 passwords that do not contain a lower case letter. Then $|A| = 36^7$
- C be the set of length 7 passwords that do not contain an number letter. Then $|A| = 52^7$

Under this set up,

- $A \cap B$ is the set of length 7 passwords that contain no upper nor lower case letter, i.e. only numbers. Then $|A \cap B| = 10^7$
- $B \cap C$ is the set of length 7 passwords that contain no lower case letter nor number, i.e. only upper case letter. Then $|B \cap C| = 26^7$
- $A \cap C$ is the set of length 7 passwords that contain no upper case letter nor number, i.e. only lower case letter. Then $|A \cap C| = 26^7$
- $A \cap B \cap C$ is the set of length 7 passwords that contain nothing. Then $|A \cap B \cap C| = 0$.
- $A \cup B \cup C$ is the set of length 7 passwords that are missing one of the three required components (i.e. the invalid passwords). Here, by inclusion-exclusion,

$$\begin{aligned} |A \cup B \cup C| &= |A| + |B| + |C| - |A \cap B| - |B \cap C| - |A \cap C| + |A \cap B \cap C| \\ &= 52^7 + 2 \times 36^7 - 2 \times 26^7 - 10^7 \end{aligned}$$

Lastly, the number of valid passwords is

$$|\mathcal{S}| - |A \cup B \cup C| = 62^7 - 52^7 - 2 \times 36^7 + 2 \times 26^7 + 10^7$$

Rubric: There are 10 items to compute, each for one point.

7. Encoding and Decoding (10 points)

In an old Atari game, you navigate your character from one point to another using the **four arrow keys: Up, Down, Left, Right**. Each move is represented by a press of any one of the four arrow keys above.

(a) (3 points) How many different ways are there to navigate your character by pressing n arrow keys? For example, if $n = 3$, one possible way to navigate your character is by pressing *Up, Up, Down.*

Answer: Each step has 4 possibilities so in total, there are 4^n different sequences of moves.

(b) (3 points) How many bits (0's and 1's) are required to represent such a way of navigating your character by pressing n arrow keys?

Answer: The optimal number of bit is given by $\lceil 4^n \rceil = 2n$

(c) (4 points) Describe how to encode a way of navigating your character by pressing n arrow keys, using the number of bits you gave in part (b).

Illustrate your description by showing how your encoding scheme would encode the following key sequence.

$$Up,\ Up,\ Down,\ Down,\ Left,\ Right,\ Left,\ Right$$

Answer: There are now different ways to encode a sequence of moves. Each move should be represented in **two** bits. Make sure that the student's output for the last part matches their encoding scheme.

8. Discrete Probability (30 points)

(a) (5 points) Assume that every time you open a digital loot-box in *Overwatch*, your chance of getting your desired item is 2/3. How many digital loot-boxes must you open so that the probability of you getting your desired item is greater than 97%?

Answer: The probability of getting your desired item when opening n boxes is

$$1 - \left(\frac{1}{3}\right)^n .$$

Now solve the following inequality for n:

$$\begin{aligned} 1 - \left(\frac{1}{3}\right)^n > 0.97 &\Rightarrow \left(\frac{1}{3}\right)^n < 0.03 = \frac{3}{100} \\ &\Rightarrow 3^{n+1} > 100 \\ &\Rightarrow n > \log_3(100) - 1 \approx 3.1 \end{aligned}$$

So the minimum number of boxes we need is 4.

Note: **Minus one point** if student leave their answer unsimplified in terms of the $\log_3$ function.

(b) (5 points) Suppose that a Bayesian spam filter is trained on a set of $10,000$ **spam messages and** 5000 **messages that are not spam**. The word "prize" appears in 800 **spam messages and in** 200 **messages that are not spam**. 50% of the incoming email messages to the inbox are spam. Find the probability that a received message containing the word "prize" is spam.

Answer: Let

- E be the event that the received message contains the word "prize"
- F be the event that the message is a spam. Then $P(F) = P(F^C) = 1/2$

$P(E|F) = \dfrac{800}{10000} = \dfrac{2}{25}$

$P(E|F^C) = \dfrac{200}{5000} = \dfrac{1}{25}$

Then by Bayes' theorem:

$$P(F|E) = \frac{P(E|F)P(F)}{P(E|F)P(F) + P(E|F^C)P(F^C)} = \frac{(2/25)(1/2)}{(2/25)(1/2) + (1/25)(1/2)} = \frac{2}{3}$$

Rubric:

- **1 point** for correctly state the Bayes' formula
- **1 point** for giving $P(F) = P(F^C) = 1/2$
- **1 point each** for computing $P(E|F)$ and $P(E|F^C)$ correctly
- **1 point** for the correct final answer

(c) (5 points) Suppose we keep rolling **a pair of fair dice** until we get a roll with a **total of 10**. What is the expected number of rolls until the ending condition is met?

Answer:

Now we let X be the random variable that keeps track of the number of rolls we need to get the desired result and let A be the event that we get a sum of 10 in the first time. Here, $P(A) = 1/12$.

We have

$$E(X) = P(A)E(X|A) + P(A^C)E(X|A^C) = \left(\frac{1}{12}\right)(1) + \frac{11}{12}(1 + E(X)) = 1 + \frac{11}{12}E(X).$$

Solving for $E(X)$ gives $E(X) = 12$.

Remark: Another method is to use the series

$$E(X) = \sum_{1}^{\infty} kP(X = k) = \sum_{1}^{\infty} k\left(\frac{1}{12}\right)\left(\frac{11}{12}\right)^{k-1} = 12.$$

Rubric:

- Only give **2 points** if student said the expectation is 12 or, in general, $1/p$ without any work to prove the result.
- Give **4 points** if student obtained the wrong probability $P(A)$ but still followed through with the above argument and showed that $E(X) = 1/p$ for whatever probability p they obtained.

(d) (5 points) Let G be a simple undirected graph with n vertices and m edges. We partition the vertices of G at random into two sets L and R, putting each vertex in L with probability 1/2, and in R with probability 1/2, (independently for each vertex). The *cut* is the set of edges with one endpoint in L and the other in R. Use **linearity of expectation** to find the expected number of edges in the cut.

Answer: Let X be the random variable that counts the number of edges in the cut. Then $X = \sum_{i=1}^{m} X_i$ where

$$X_i = \begin{cases} 1 & \text{if the m-th edge is in the cut} \\ 0 & \text{otherwise} \end{cases}$$

Here, the probability that the m-th edge is in the cut is the same as the probability that its two end points belong to different sets L and R. This probability is 1/2 which means $E(X_i) = P(X_i = 1) = 1/2$ for each $1 \leq i \leq m$.
Hence, $E(X) = m/2$

(e) There is no partial credit for the following problems, so you do not need to show any work. You may leave your answers as unsimplified expressions with factorials, exponents, binomial coefficients, etc.

The following problems refer to a standard deck of 52 cards with 13 cards in each of four suits (heart, diamond, club, and space), and no jokers nor wild cards. You are dealt a random hand of **five** cards from this standard deck of cards.

(e.1) (2 points) How many 5-card hands are possible?

Answer: $\binom{52}{5}$

(e.2) (2 points) What is the probability that all the cards your hand are hearts?

Answer: $\binom{13}{5} \Big/ \binom{52}{5}$

(e.3) (2 points) What is the probability that your hand contains no hearts?

$\binom{39}{5} \Big/ \binom{52}{5}$

(e.4) (2 points) What is the probability that your hand contains exactly 2 hearts?

$\binom{13}{2} \times \binom{39}{3} \Big/ \binom{52}{5}$

(e.5) (2 points) What is the probability that your hand has all four suits present?

$4 \times 13^3 \times \binom{13}{2} \Big/ \binom{52}{5}$