Assignment 1 – Discrete Review

Answer the questions in the boxes provided on the question sheets. If you run out of room for an answer, add a page to the end of the document.

Related Readings: `http://pages.cs.wisc.edu/~hasti/cs240/readings/`

Name: ______________________ Wisc id: ______________________

Logic

1. Using a truth table, show the equivalence of the following statements.

(a) $P \vee (\neg P \wedge Q) \equiv P \vee Q$

Solution:

P	Q	$\neg P$	$\neg P \wedge Q$	$P \vee (\neg P \wedge Q)$	$P \vee Q$
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

(b) $\neg P \vee \neg Q \equiv \neg(P \wedge Q)$

Solution:

P	Q	$\neg P$	$\neg Q$	$\neg P \vee \neg Q$	$P \wedge Q$	$\neg(P \wedge Q)$
0	0	1	1	1	0	1
0	1	1	0	1	0	1
1	0	0	1	1	0	1
1	1	0	0	0	1	0

(c) $\neg P \vee P \equiv \text{true}$

Solution:

P	$\neg P$	$\neg P \vee P$
0	1	1
0	1	1
1	0	1
1	0	1

(d) $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$

Solution:

P	Q	R	$Q \wedge R$	$P \vee (Q \wedge R)$	$P \vee Q$	$P \vee R$	$(P \vee Q) \wedge (P \vee R)$
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Sets

2. Based on the definitions of the sets A and B, calculate the following: $|A|$, $|B|$, $A \cup B$, $A \cap B$, $A \setminus B$, $B \setminus A$.

   (a) $A = \{1, 2, 6, 10\}$ and $B = \{2, 4, 9, 10\}$

   Solution:

   $$\begin{aligned} |A| &= 4 \\ |B| &= 4 \\ A \cup B &= \{1, 2, 4, 6, 9, 10\} \\ A \cap B &= \{2, 10\} \\ A \setminus B &= \{1, 6\} \\ B \setminus A &= \{4, 9\} \end{aligned}$$

   (b) $A = \{x \mid x \in \mathbb{N}\}$ and $B = \{x \in \mathbb{N} \mid x \text{ is even}\}$

   Solution:

   $$\begin{aligned} |A| &= \infty \\ |B| &= \infty \\ A \cup B &= \{x \mid x \in \mathbb{N}\} \\ A \cap B &= \{x \in \mathbb{N} \mid x \text{ is even}\} \\ A \setminus B &= \{x \in \mathbb{N} \mid x \text{ is odd}\} \\ B \setminus A &= \{\} \end{aligned}$$

Relations and Functions

3. For each of the following relations, indicate if it is reflexive, antireflexive, symmetric, antisymmetric, or transitive.

   (a) $\{(x, y) : x \leq y\}$

   Solution: reflexive, antisymmetric, transitive

   (b) $\{(x, y) : x > y\}$

   Solution: antireflexive, antisymmetric, transitive

(c) $\{(x, y) : x < y\}$

Solution: antireflexive, antisymmetric, transitive

(d) $\{(x, y) : x = y\}$

Solution: reflexive, symmetric, antisymmetric, transitive

4. For each of the following functions (assume that they are all $f : \mathbb{Z} \to \mathbb{Z}$), indicate if it is surjective (onto), injective (one-to-one), or bijective.

(a) $f(x) = x$

Solution: bijection

(b) $f(x) = 2x - 3$

Solution: injective

(c) $f(x) = x^2$

Solution: None. Not surjective because the image does not cover the entire co-domain.

5. Show that $h(x) = g(f(x))$ is a bijection if $g(x)$ and $f(x)$ are bijections.

Solution: Since f and g are bijections, they are injective. This implies that the composition of g and f will produce a unique value for every x and, hence, be injective. Moreover, since g and f are bijections, the size of the domains and ranges must be same, implying that their composition (which produces a unique value for every x) is also surjective. $\square$

Induction

6. Prove the following by induction.

(a) $\sum_{i=1}^{n} i = n(n+1)/2$

Solution:
Base case: $n = 1$; $n(n+1)/2 = 1$
Induction step: By ind hyp, $\sum_{i=1}^{k} i = k(k+1)/2$. For $k+1$,

$$k(k+1)/2 + k + 1 = (k(k+1) + 2k + 2)/2 = (k+1)(k+2)/2 \; .$$

□

(b) $\sum_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$

Solution:
Base case: $n = 1$; $n(n+1)(2n+1)/6 = 1$
Induction step: By ind hyp, $\sum_{i=1}^{k} i^2 = k(k+1)(2k+1)/6$. For $k+1$,

$$\begin{aligned} k(k+1)(2k+1)/6 + (k+1)^2 &= (k(k+1)(2k+1) + 6(k+1)^2)/6 \\ &= (k+1)(2k^2 + 7k + 6)/6 \\ &= (k+1)(k+2)(2k+3)/6 \; . \end{aligned}$$

□

(c) $\sum_{i=1}^{n} i^3 = n^2(n+1)^2/4$

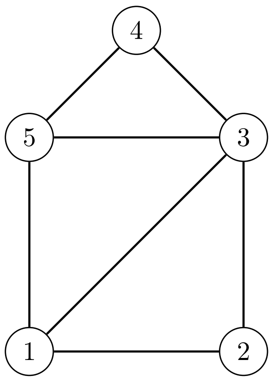
Solution:
Base case: $n = 1$; $n^2(n+1)^2/4 = 1$
Induction step: By ind hyp, $\sum_{i=1}^{k} i^3 = k^2(k+1)^2/4$. For $k+1$,

$$\begin{aligned} k^2(k+1)^2/4 + (k+1)^3 &= (k^2(k+1)^2 + 4(k+1)^3)/4 \\ &= (k+1)^2(k^2 + 4k + 4)/4 \\ &= (k+1)^2(k+2)^2/4 \; . \end{aligned}$$

□

Graphs and Trees

7. Give the adjacency matrix, adjacency list, edge list, and incidence matrix for the following graph.



Solution:
Adjacency Matrix:

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Adjacency List:

$$\begin{pmatrix} (2,3,5) \\ (1,3) \\ (1,2,4,5) \\ (3,5) \\ (1,3,4) \end{pmatrix}$$

Edge List:
$\big((1,2),(1,3),(1,5),(2,3),(3,4),(3,5),(4,5)\big)$

Incidence Matrix:

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

8. How many edges are there is a complete graph of size n? Prove by induction.

Solution:
There are $h(n) = n(n-1)/2$ edges.

By induction on the number of nodes. Base case: $n = 1$. $h(1) = 0$.

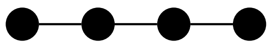
Induction step: By ind hyp, k_i has $i(i-1)/2$ edges. Adding a node for k_{i+1} adds i edges. Hence,

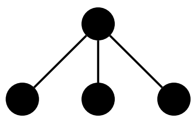
$$h(i+1) = i(i-1)/2 + i = \frac{i(i-1)+2i}{2} = \frac{i((i-1)+2)}{2} = (i+1)i/2 \; .$$

$\square$

9. Draw all possible (unlabelled) trees with 4 nodes.

Solution:





10. Show by induction that, for all trees, $|E| = |V| - 1$.

Solution:

By (structural) induction on the number of nodes. Base Case: Single node; no edges

Induction step: By ind hyp, a tree of k nodes has $k-1$ edges. For any tree of size k and any possible way to add a node and remain a tree, we can only ever add a single edge otherwise we'd introduce a cycle. It should be noted that this covers all possible trees of with $k+1$ nodes. Hence,

$$|E| = k - 1 + 1 = k = |V| - 1 \ .$$

$\square$

Counting

11. How many 3 digit pin codes are there?

> **Solution:** For each digit, there are 10 possibilities, so 10^3.

12. What is the expression for the sum of the ith line (indexing starts at 1) of the following:

1
2 3
4 5 6
7 8 9 10
$\vdots$

> **Solution:**
> Find the last value in line $i-1$. We call this value x.
>
> $$x = \sum_{j=1}^{i-1} j = \frac{i(i+1)}{2} - i = \frac{i(i-1)}{2}$$
>
> Find sum of line i. Note that line i contains i numbers.
>
> $$.\sum_{k=1}^{i}(x+k) = ix + \frac{i(i+1)}{2} = \frac{i^2(i-1)}{2} + \frac{i(i+1)}{2} = \frac{i^3+i}{2}$$

13. A standard deck of 52 cards has 4 suits, and each suit has card number 1 (ace) to 10, a jack, a queen, and a king. A standard poker hand has 5 cards. For the following, how many ways can the described hand be drawn from a standard deck.

(a) A royal flush: all 5 cards have the same suit and are 10, jack, queen, king, ace.

> **Solution:** 4 (one for each suit)

(b) A straight flush: all 5 cards have the same suit and are in sequence, but not a royal flush.

> **Solution:** $4 * 10 - 4 = 36$ (4 suits; 10 possible starting positions less the 4 royal flushes)

(c) A flush: all 5 cards have the same suit, but not a royal or straight flush.

> **Solution:** $4 * {}_{13}C_5 - 40 = 5108$ (4 suits; choosing 5 from 13 options less the 40 straight and royal flushes)

(d) Only one pair (2 of the 5 cards have the same number/rank, while the remaining 3 cards all have different numbers/ranks):

> **Solution:** $13 * {}_4C_2 * {}_{12}C_3 * 4^3 = 1,098,240$ (13 possible values for the pair of 4 choose 2 suits; 12 choose 3 values for the other cards each with 4 possible suits)

Proofs

14. Show that $2x$ is even for all $x \in \mathbb{N}$.

(a) By direct proof.

Solution: By definition, an even number is divisible by 2, and $2x$ is divisible by 2. $\square$

(b) By contradiction.

Solution: Assume to the contrary that $2x$ is odd. This implies that $2x$ is not perfectly divisible by 2. However, $2x/2 = x$, a contradiction.

15. For all $x, y \in \mathbb{R}$, show that $|x + y| \leq |x| + |y|$. (Hint: use proof by cases.)

Solution:

Case 1: $x \geq 0, y \geq 0$. $|x + y| = x + y = |x| + |y|$.

Case 2: $x < 0, y < 0$. $|x + y| = -(x + y) = |x| + |y|$.

Case 3: $x = 0, y < 0$. $|x + y| = |y| < |x| + |y|$.

Case 4: $x > 0, y < 0$ **and** $|x| \geq |y|$. $|x + y| < |x| \leq |x| + |y|$.

Case 5: $x > 0, y < 0$ **and** $|x| < |y|$. $|x + y| < |y| \leq |x| + |y|$.

WLOG: Case 3 covers $x < 0, y = 0$; Case 4 covers $y > 0, x < 0$ and $|y| \geq |x|$; and Case 5 covers $y > 0, x < 0$ and $|y| < |x|$

Program Correctness (and Invariants)

16. For the following algorithms, describe the loop invariant(s) and prove that they are sound and complete.

(a)

Algorithm 1: findMin

```
Input: a: A non-empty array of integers (indexed starting at 1)
Output: The smallest element in the array
begin
    min ← ∞
    for i ← 1 to len(a) do
        if a[i] < min then
            min ← a[i]
        end
    end
    return min
end
```

Solution:

Invariant: At the end of step i of the loop, min contains the minimum value from index 1 to i of a.

[Proof: By induction on the iterations. After the first iteration, min is set to $a[1]$. Assume that min is minimum value of $a[1, \ldots, k]$ after the kth iteration. After the $(k+1)$-st iteration, min will be the minimum of min and $a[k+1]$. Hence, min is the minimum of $a[1, \ldots, k+1]$.]

Soundness: From the loop invariant, after completing the step equal to the length of the input array, min will contain the minimum element.

Completeness: With each iteration of the loop, the counter i moves one step closer to the end of the input array, and the loop (and the algorithm) terminates after reaching the end of the array.

(b)

Algorithm 2: InsertionSort

Input: a: A non-empty array of integers (indexed starting at 1)
Output: a sorted from largest to smallest

```
begin
    for i ← 2 to len(a) do
        val ← a[i]
        for j ← 1 to i − 1 do
            if val > a[j] then
                shift a[j..i − 1] to a[j + 1..i]
                a[j] ← val
                break
            end
        end
    end
    return a
end
```

Solution:

Invariant: At the end of step i of the outer loop, the items $a[1..i]$ are sorted from largest to smallest.

[Proof: By induction on the iterations. After the first iteration, the item $a[1]$ is sorted. Assume that $a[1, \ldots, k]$ is sorted after the kth iteration. After the $(k+1)$-st iteration, $a[k+1]$ will be placed at $a[j]$ after the sorted $a[1, \ldots, j-1]$ items larger than $a[k+1]$. The sorted $a[j, k]$ items less than $a[k+1]$ are shifted to $a[j+1, k+1]$. Hence, $a[1, \ldots, k+1]$ is sorted.]

Soundness: From the loop invariant, after completing the step (of the outer loop) equal to the length of the input array, a will be sorted from largest to smallest.

Completeness: With each iteration of the loop, the outer loop counter i moves one step closer to the end of the input array, and the outer loop (and the algorithm) terminates after reaching the end of the array. For each iteration, the inner loop increments by 1 from 1 to i (the counter of the outer loop), and will always terminate.

Recurrences

17. Solve the following recurrences.

(a) $c_0 = 1; c_n = c_{n-1} + 4$

Solution:

$$\begin{aligned} c_n &= c_{n-1} + 4 \\ &= c_{n-2} + 4 + 4 \\ &= c_0 + 4n \\ &= 4n + 1 \end{aligned}$$

(b) $d_0 = 4; d_n = 3 \cdot d_{n-1}$

Solution:

$$\begin{aligned} d_n &= 3 \cdot d_{n-1} \\ &= 3 \cdot 3 \cdot d_{n-2} \\ &= 3^n d_0 \\ &= 4 \cdot 3^n \end{aligned}$$

(c) $T(1) = 1; T(n) = 2T(n/2) + n$ (An upper bound is sufficient.)

Solution:

Unrolling

- Let k be the depth of recursion
- At base case: $k = log_2(n)$

$$\begin{aligned} T(n) &= 2T(\frac{n}{2}) + n \\ &= 2^k T(\frac{n}{2^k}) + kn \\ &= 2^{log_2(n)} T(1) + log_2(n) * n \\ &= n + nlog_2(n) \\ &= O(nlog_2 n) \end{aligned}$$

Recurrence Tree

- Let k be the current layer of the tree (root node is in layer 0)
- *Input size tree:* Root node has size n, recursively splits into 2 children nodes each with size $\frac{n}{2^k}$
- *Work tree:* Identical to input size tree
- Height of tree (using base case): $\frac{n}{2^h} = 1$, $h = log_2(n)$
- Sum of each level of work tree: n

$$\begin{aligned} T(n) &= \text{work not at leaves} + \text{work at leaves} \\ &= \sum_{k=0}^{h-1} n + 2^h * O(1) \\ &= \sum_{k=0}^{log_2(n)-1} n + 2^{log_2(n)} * O(1) \\ &= nlog_2(n) + n \\ &= O(nlog_2 n) \end{aligned}$$

(d) $f(1) = 1; f(n) = \sum_1^{n-1} (i \cdot f(i))$
(Hint: compute $f(n+1) - f(n)$ for $n > 1$)

Solution:

$$\begin{aligned} f(n+1) - f(n) &= \sum_1^{n} (i \cdot f(i)) - \sum_1^{n-1} (i \cdot f(i)) \\ &= (n) \cdot f(n) \\ f(n+1) &= (n+1) f(n) \end{aligned}$$

Which holds for $n + 1 > 1$. Hence,

$$\begin{aligned} f(n) &= n \cdot f(n-1), \text{ for } n > 2 \\ &= n \cdot n - 1 \cdots 3 \cdot f(2) \\ &= \frac{n!}{2} \cdot 1 \cdot f(1) \\ &= \frac{n!}{2} \end{aligned}$$

Note that $f(2) = 1 \cdot f(1) = 1 = 2!/2$. Therefore,

$$f(n) = \begin{cases} 1, \text{ when } n = 1 \\ \frac{n!}{2}, \text{ otherwise} \end{cases}$$

Coding Question: Hello World

Most assignments will have a coding question. You can code in C, C++, C#, Java, Python, or Rust. You will submit a Makefile and a source code file.

Makefile: In the Makefile, there needs to be a build command and a run command. Below is a sample Makefile for a C++ program. You will find this Makefile in assignment details. Download the sample Makefile and edit it for your chosen programming language and code.

```
#Build commands to copy:
#Replace g++ -o HelloWorld HelloWord.cpp below with the appropriate command.
#Java:
#       javac source_file.java
#Python:
#       echo "Nothing to compile."
#C#:
#       mcs -out:exec_name source_file.cs
#C:
#       gcc -o exec_name source_file.c
#C++:
#       g++ -o exec_name source_file.cpp
#Rust:
#       rustc source_file.rs

build:
        g++ -o HelloWorld HelloWord.cpp

#Run commands to copy:
#Replace ./HelloWorld below with the appropriate command.
#Java:
#       java source_file
#Python 3:
#       python3 source_file.py
#C#:
#       mono exec_name
#C/C++:
#       ./exec_name
#Rust:
#       ./source_file

run:
        ./HelloWorld
```

18. **HelloWorld Program Details**

    The input will start with a positive integer, giving the number of instances that follow. For each instance, there will be a string. For each string s, the program should output Hello, s! on its own line.

    A sample input is the following:

```
3
World
Marc
Owen
```

The output for the sample input should be the following:

```
Hello, World!
Hello, Marc!
Hello, Owen!
```