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: Ki Min Kang

Wisc id: 908 404 5062

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

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

Solution:

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

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

Solution:

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

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

Solution:

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

$|A| = 4$, $|B| = 4$

$A \cup B = \{1, 2, 4, 6, 9, 10\}$

$A \cap B = \{2, 10\}$

$A \setminus B = \{1, 6\}$

$B \setminus A = \{4, 9\}$

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

Solution:

$|A| = \infty$, $|B| = \infty$

$A \cup B = \{x \mid x \in N\}$

$A \cap B = \{x \in N \mid x \text{ is even}\}$

$A \setminus B = \{x \in N \mid x \text{ is odd}\}$

$B \setminus A = \{\}$

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: injection

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

Solution: None, since the function does not cover the entire co-domain, $\mathbb{Z}$.

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

Solution:

When we have two functions, $f(x)$ and $g(x)$, which are both bijections, this indicates that each of them is injective and surjective.

Now, we will show h is a bijection, given that f and g are bijection.

① Since f and g are injective, each element in the domain of f maps to a unique element in the range of f, and the same goes for g. When we compose two injective functions, the result is also injective. Therefore, h is injective.

② Since f and g are surjective, every element in the range of f and g is covered by elements from each of their domain. So, every element in the range of g is from the domain of f. Therefore, h is surjective.

In summary, ①, ② show that h inherits the injective and surjective properties from f and g. Therefore, h is bijection.

Induction

6. Prove the following by induction.

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

Solution:

Base case : $n=1$

$n(n+1)/2 = 1$

The base case holds.

Inductive Step : Assume that the formula holds for $n=k$, so we have

$\sum_{i=1}^{k} i = \frac{k(k+1)}{2}$

Now, we will show it holds for $n=k+1$

$\sum_{i=1}^{k+1} i = \sum_{i=1}^{k} i + (k+1)$

$= \frac{k(k+1)}{2} + (k+1)$

$= \frac{(k+1)((k+1)+1)}{2}$

Thus, the formula holds for $n=k+1$, which indicates it holds for all natural numbers n.

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

Solution:

Base Case : $n=1$

$\sum_{i=1}^{n} i^2 = \frac{1 \cdot 2 \cdot 3}{6} = 1$

The base case holds.

Inductive Step : Assume that the formula holds for $n=k$, so we have

$\sum_{i=1}^{k} i^2 = \frac{k(k+1)(2k+1)}{6}$

Now, we will show it holds for $n=k+1$

$\sum_{i=1}^{k+1} i^2 = i^2 + (k+1)^2$

$= \frac{k(k+1)(2k+1)}{6} + (k+1)^2$

$= \frac{k(k+1)(2k+1) + 6(k+1)^2}{6} = \frac{(k+1)((k+1)+1)(2(k+1)+1)}{6}$

Thus, the formula holds for $n=k+1$, and by induction, it holds for all natural number n.

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

Solution:

Base case : $n=1$

$\sum_{i=1}^{1} i^3 = 1 = \frac{1^2(2)^2}{4} = 1$

The base case holds.

Inductive Step : Assume that the formula holds for $n=k$, so we have

$\sum_{i=1}^{k} i^3 = \left(\frac{k(k+1)}{2}\right)^2$

Now, we will show it holds for $n=k+1$

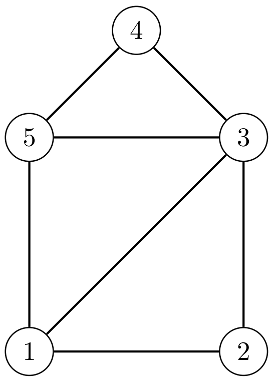
$\sum_{i=1}^{k+1} i^3 = \left(\frac{k(k+1)}{2}\right)^2 + (k+1)^3$

$= \frac{k^2(k+1)^2 + 4(k+1)^3}{4} = \left(\frac{(k+1)((k+1)+1}{2}\right)^2$

Thus, the formula holds for $n=k+1$, and by induction, it holds for all natural numbers n.

Graphs and Trees

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



Solution:

Adjacency Matrix

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

Adjacency List:

((2,3,5), (1,3), (1,2,4,5), (3,5), (1,3,4))

Edge List

((1,2), (1,3), (1,5), (2,3), (3,4), (3,5), (4,5))

Incidence Matrix

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

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

Solution:

The number of edges $h(n)$ in a complete graph of size n can be expressed as:

$h(n) = n(n-1)/2$

- Base case

  $n=1$

  $h(1) = 0$, so the base case holds

- Inductive step:

  Assume that the formula holds for some positive integer K:

  $h(K) = \frac{K(K+1)}{2}$

  Now, we will show the formula holds for $K+1$ vertices.

  When we add one more vertex to a complete graph with K vertices, this new vertex will be connected to other all existing K vertices, adding K new edges.

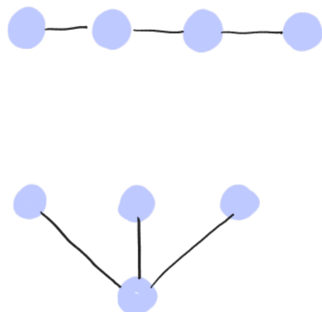
  Thus, the number of edges in the graph with $K+1$ vertices can be expressed as:

$$\begin{aligned} h(K+1) &= h(K) + K \\ &= \frac{K(K-1)}{2} + K \\ &= \frac{K(K+1)}{2} \end{aligned}$$

Thus, the formula holds for $n = K+1$, and by induction, it holds for all natural numbers n.

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

Solution:



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

Solution:

Base Case : Consider a tree with a single node $(n=1)$
Since there is no other nodes connected,
this tree has no edge.
$|E| = |V| - 1 = 1 - 1 = 0$
The base case holds

Inductive Step : Assume that the formula holds for a tree with K nodes :
The number of edges is $K-1$

Now we will add one more node to make it $K+1$ nodes
Then, we add only one more edge to connect this new node to the tree
to prevent from creating a cycle

Thus, the total number of edges for a tree with $K+1$ nodes is K.

$|E| = (K-1) + 1 = K = |V| - 1$

Therefore, the formula holds for $n=K+1$, and by induction, it holds for all natural numbers n.

Counting

11. How many 3 digit pin codes are there?

Solution: Each digit has 10 possibilities.
Therefore, there are possible 10^3 codes.

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
⋮

Solution:
First, we will find the last value in line $i-1$, denoted by x.
We will find x by using a well-known formula:
$x = \sum_{a=1}^{i-1} a = \frac{i(i-1)}{2}$

Then, we will find the sum of line i.
The first number in line i will be $x+1$, since it comes right after x.
The sum of line i can be found by adding x to each numer in line i:
$\sum_{k=1}^{i}(x+k) = ix + \sum_{k=1}^{i} k = i\left(\frac{i(i-1)}{2}\right) + \frac{i(i+1)}{2}$
$= \frac{i^3+i}{2}$

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

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

Solution: $4 \times (10-1) = 36$

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

Solution: $(4 \times {}_{13}C_5) - 40 = 5108$

(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: ${}_4C_2 \times 13 \times {}_{12}C_3 \times 4 \times 4 \times 4$

Proofs

14. Show that $2x$ is even for all $x \in \mathbb{N}$.
    (a) By direct proof.

Solution: Every even numbers are divisible by 2 by definition.
Since $2x$ is divisible by 2, the statement is true

(b) By contradiction.

Solution: Let's say that $2x$ is some odd number.
This indicates that $2x$ cannot be divided by 2.
However, $2x$ can be divided by 2 ($2x/2 = x$)
Therefore, the statement is true by 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) $x > 0, y < 0$ ($|x| > |y|$) : $|x+y| < |x| \leq |x|+|y|$ $\Rightarrow$ This covers $y > 0, x < 0$ ($|y| \geq |x|$)

case 4) $x > 0, y < 0$ ($|x| < |y|$) : $|x+y| < |y| \leq |x|+|y|$ $\Rightarrow$ This covers $y > 0, x < 0$ ($|y| \leq |x|$)

Program Correctness (and Invariants)

알고리즘 정확성 증명 알고리즘 정확성 알고리즘 완전성

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 :
  The variable 'min' will contain the minimum value from index 1 to i after the loop.
  Proof) Base case : After the first iteration, 'min' is set to 'a[1]', which is the smallest value from index 1 to 1.
  Inductive Step : Assume that 'min' is the minimum value of a[1] ⋯ a[K] after K iteration.
  After the (K+1) iteration, 'min' will be the minimum value of previous 'min' and 'a[K+1]'.
  'min' will be the minimum of 'a[1] ⋯ a[K+1]'.
  Therefore, the formula holds for n=K+1, and by induction, it holds for all natural numbers n.

- Soundness : The loop invariant above ensures that after the step equal to the length of the array will set 'min' variable to the mininumum element from the array.

- Completeness : The loop increases the counter i by one in each iteration, which will eventually traverse every element of the array and find the minimum element 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 $i \leftarrow 2$ **to** $len(a)$ **do**
    $val \leftarrow a[i]$
    for $j \leftarrow 1$ **to** $i-1$ **do**
      if $val > a[j]$ **then**
        shift $a[j..i-1]$ to $a[j+1..i]$
        $a[j] \leftarrow val$
        break
      end
    end
  end
  return a
end

Solution:

- Invariant:
  The sub array 'a[i ... 1]' will be sorted from largest to smallest after the end of the outer loop.

  Proof) Base case: 'a[1]' contains only one element and therefore it is trivially sorted.

  Inductive step: Assume that after the 'k'th iteration, the sub array 'a[1...k]' is sorted. During the 'k+1'th iteration, the algorithm places 'a[k+1]' in the correct position relative to the already sorted 'a[1...k]', and the sub-array 'a[1... k+1]' will be sorted after the '(k+1)'th iteration.

  Therefore, the formula holds for n=k+1, and by induction, it holds for all natural numbers n.

- Soundness: The loop invariant above ensures that after completing the outer loop for all elements in the array, the entire array 'a' will be sorted from largest to smallest.

- Completeness: The outer-loop increases the counter i by one in each iteration, which will eventually traverse every element of the array. The algorithm ends after the last element has been placed in the correct position. Each iteration of the inner loop ensures to terminate since it increments from '1' to 'i' by 1, which will always terminate.

Recurrences

17. Solve the following recurrences. Show work and do not use the master theorem.

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

Solution:

$c_0 = 1$, $c_n = c_{n-1} + 4$

$c_n = c_{n-1} + 4$

$= (c_{n-2} + 4) + 4$

This demonstrates each step adds the number 4.

$= c_0 + 4n$

$= 4n + 1$

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

Solution:

$d_0 = 4$, $d_n = 3 \cdot d_{n-1}$

$d_n = 3 \cdot d_{n-1}$

$= 3 \cdot (3 \cdot d_{n-2})$

this demonstrates that each step multiplies the number 3.

$= 3^n \cdot d_0$

$= 4 \cdot 3^n$

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

Solution:

$T(1) = 1,\ T(n) = 2T(n/2) + n$

The formula T represents algorithm, where problem of size 'n' is divided into two smaller problems of size 'n/2' with an additional 'n'

Let 'K' be the depth of recursion

In the base case, $K = \log_2 n$

Then, we can express T(n) as:

$$\begin{aligned} T(n) &= 2T(\tfrac{n}{2}) + n \\ &= 2^K T(\tfrac{n}{2^K}) + Kn \\ &= 2^{\log_2 n} T(1) + (\log_2 n \times n) \\ &= n + n\log_2(n) \\ &= O(n \log_2 n) \end{aligned}$$

The solution above shows that after 'K' levels of recursion, the work done is '$O(n \log_2(n))$'

(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:

$f(1) = 1,\ f(n) = \sum_1^{n-1} (i \cdot f(i))$

$f(n+1) - f(n)$ can be expressed as:

$$\sum_1^{n} (i \cdot f(i)) - \sum_1^{n-1} (i \cdot f(i))$$

$$= n \cdot f(n) + \cancel{(n-1) f(n-1)} - \cancel{(n-1) f(n-1)} + \cdots - \cancel{f(1)}$$

$$= n \cdot f(n)$$

From this we can deduce that:

$f(n+1) = (n+1) f(n)$, and this holds true for all '$n>1$'.

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

Therefore, we can now conclude that:

$$f(n) = \begin{cases} 1 & (n=1) \\ \frac{n!}{2} & (n>1) \end{cases}$$

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!
```