COMP 9020 – Additional assignment

0. **Quiz X** True or false:

   (a) For any $m, n \geq 1$: $\kappa(K_{m,n}) = \chi(K_{m,n})$ (the clique number of $K_{m,n}$ is equal to the chromatic number of $K_{m,n}$).

   (b) $(p \wedge \neg r), (q \rightarrow \neg p) \models (r \vee q)$

   (c) There are exactly three clauses in a minimal DNF of $(\neg y \vee (z \wedge (x \vee y)))$

   (d) Suppose $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ is defined recursively as follows for all $n, m \in \mathbb{N}$:

   $$f(0, m) = 0; \qquad f(n+1, m) = m + f(n, m).$$

   True or false: $f(m, n) = f(n, m)$ for all $m, n \in \mathbb{N}$.

   (e) Suppose $T(n)$ is defined recursively as:

   $$T(0) = 1; \qquad T(n) = 3T(n/3) + O(n)$$

   True or false: $T(n) \in O(n)$

   (f) Suppose $T(n)$ is defined recursively as:

   $$T(0) = 1; \qquad T(n) = 3T(n-3) + O(n)$$

   True or false: $T(n) \in O(2^n)$

   (g) There are $\binom{26}{3} . \binom{49}{2}$ ways of choosing a hand of 5 cards from a deck of 52 cards (26 red and 26 black) with at least 3 red cards.

   (h) $\sum_{k=0}^{n} \binom{n}{k} = 2^n$

   (i) $P(A) = P(B)$ if and only if $P(B|A) = P(A|B)$.

   (j) Let X be random variable representing the maximum value of two six-sided dice. Then $E[X]$, the expected value of X, is greater than or equal to 4.5.

Solution:

(a) True. Both are equal to 2.

(b) False. If $p = q = r = \mathsf{false}$ then both formulas on the left are satisfied, but the formula on the right is not.

(c) False. The formula is equivalent to $\neg y \vee z$ which is a DNF with two clauses.

(d) True. $f(m, n) = mn$ (which is provable by induction).

(e) False. By the Master theorem $T(n) \in O(n \log n)$

(f) True. $T(n) \in O((\sqrt[3]{3})^n)$ (which can be proven by induction) and as $\sqrt[3]{3} < 2$ we have $T(n) \in O(2^n)$

(g) False. This overcounts hands with more than 3 red cards. The correct number should be $\binom{26}{3}\binom{26}{2} + \binom{26}{4}\binom{26}{1} + \binom{26}{5}$

(h) True. This can be seen by expanding $2^n = (1 + 1)^n$.

(i) False. If A and B are disjoint, but $P(A) \neq P(B)$ then $P(A|B) = 0 = P(B|A)$.

(j) False. The probability that $X = k$ is $\frac{1}{6}.\frac{k-1}{6} + \frac{k-1}{6}.\frac{1}{6} + \frac{1}{36} = \frac{2k-1}{36}$ (one dice rolls k, the other rolls something less than k; or the other way around; or both roll k). Therefore the expected value of X is

$$1.\frac{1}{36} + 2.\frac{3}{36} + 3.\frac{5}{36} + 4.\frac{7}{36} + 5.\frac{9}{36} + 6.\frac{11}{36} = \frac{1 + 6 + 15 + 28 + 45 + 66}{36} = \frac{161}{36} < 4.5$$

Note: For the remainder of the assignment, *how* you arrived at your solution is as important (if not more so) than the solution itself and will be assessed accordingly. There may be more than one way to find a solution, and your approach should contain enough detail to justify its correctness. Lecture content can be assumed to be common knowledge.

1. Consider the following algorithm which finds the index (in $[i, j)$) of an item x in a sorted list L (or returns the location that x should be inserted):

Algorithm 1 BinarySearch(L, x, i, j)

```
if i = j return i
else:
  k = (i + j)/2
  if x = L[k]:
    return k
  elif x > L[k]:
    return BinarySearch(L, x, k, j)
  else:
    return BinarySearch(L, x, i, k)
```

(a) Give an asymptotic recurrence for the running time $T_B(n)$ of BinarySearch$(L, x, 0, n)$.

(b) Solving the recurrence, give an upper bound for the running time $T_B(n)$.

Solution: For the base case (the if part) we have $T_B(0) = O(1)$. For the recursive case (the else part) we have [going line by line]

$$T_B(n) = O(1) + O(1) + O(1) + T_B(n/2) = O(1) + T_B(n/2).$$

This falls into Case 2 of the Master Theorem and can be resolved as $T_B(n) \in O(\log n)$.

Now consider the following algorithm for sorting a list:

Algorithm 2 InsertionSort(L)

```
// Recursively call InsertionSort on all but the last element
Let L' = InsertionSort(L[0 : n − 1])
// Find the index of L[n] in the sorted list L'
Let i = BinarySearch(L', L[n], 0, n − 1)
// Insert L[n] into L'
return L'[0 : i] + L[n] + L'[i : n − 1]
```

(c) Give an asymptotic recurrence for the running time $T_I(n)$, of InsertionSort(L) when L is a list of length n. *Hint: Your answer should include T_B.*

(d) Solving the recurrence, give an upper bound for the running time $T_I(n)$.

> **Solution:** Going line by line and using part (b) we have:
>
> $$T_I(n) = T_I(n-1) + T_B(n-1) + O(1) = T_I(n-1) + O(\log n).$$
>
> From the lectures this can be resolved as $T_I(n) \in O(n \log n)$.

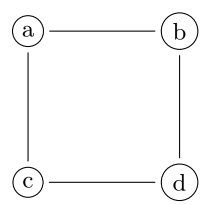
(20 marks)

2. Consider the following algorithm for navigating a graph $G = (V, E)$:

Algorithm 3 DFS(V, E, v, X)

```
1: Add v to X
2: For all vertices w with {v, w} ∈ E:
3:    if w ∉ X:
4:       DFS(V, E, w, X)
```

(a) Run the algorithm DFS(V, E, a, X) where initially $X = \emptyset$ and (V, E) is the following graph:



Assume the vertices in line 2 of the algorithm are considered in alphabetic order. What order are the vertices added to X?

(b) Give an asymptotic recurrence for the running time $T(n)$ when it is run on a graph with n vertices.

(c) Solving the recurrence, give an upper bound for the running time $T(n)$.

Solution:

(a) The vertices will be added to X in the following order: a, b, d, c.

(b) We note that the recursive call only gets made on vertices not in X, so as we add vertices to X the number of recursive calls decreases (and the algorithm will terminate). A simple analysis is that, in the worst case, the for loop will run $n-1$ times and in each loop will call DFS recursively on (because v is in X) what is effectively a graph with $n-1$ vertices. Other operations are constant time, so we have:

$$T(n) = O(1) + (n-1) \times T(n-1).$$

(c) So $T(n) \in O(n!)$. Note A more detailed analysis observes that in the entirety of a run of the algorithm, each edge is used in the second line exactly once, so the running time will be $O(|E|) = O(n^2)$.

(10 marks)

3. (a) How many well-formed formulas can be constructed from one $\vee$; one $\wedge$; two parenthesis pairs (,); and the three literals p, $\neg p$, and q?

   (b) Under the equivalence relation defined by **logical equivalence**, how many equivalence classes do the formulas in part (a) form?

Solution:

(a) We will count the number of well-formed formulas that use all symbols exactly once. We note that the parentheses are tied to the operations $\wedge$ and $\vee$ and there are two "shapes" of formula: $(l_1 op_1 (l_2 op_2 l_3))$ and $((l_2 op_2 l_3) op_1 l_1)$. There are $2 \times 1 = 2$ choices for op_1, op_2. There are $3 \times 2 \times 1 = 6$ choices for l_1, l_2, l_3. Therefore, there are $2.2.6 = 24$ formulas in total.

(b) We note that since $(\varphi \vee \psi)$ is logically equivalent to $(\psi \vee \varphi)$ and $(\varphi \wedge \psi)$ is logically equivalent to $(\psi \wedge \varphi)$ we can reduce the 24 formulas from above to the following six (possibly not distinct) classes:

$I.$	$(p \vee (\neg p \wedge q))$	$II.$	$(\neg p \vee (p \wedge q))$	$III.$	$(q \vee (p \wedge \neg p))$
$IV.$	$(p \wedge (\neg p \vee q))$	$V.$	$(\neg p \wedge (p \vee q))$	$VI.$	$(q \wedge (p \vee \neg p))$

Since

$$(q \vee (p \wedge \neg p)) \equiv (q \vee \bot) \equiv q \equiv (q \wedge \top) \equiv (q \wedge (p \vee \neg p))$$

we see that III and VI are the same class.

For the other cases we have:

I $(p \vee (\neg p \wedge q)) \equiv ((p \vee \neg p) \wedge (p \vee q)) \equiv (\top \wedge (p \vee q)) \equiv (p \vee q)$

II $(\neg p \vee (p \wedge q)) \equiv ((\neg p \vee p) \wedge (\neg p \vee q)) \equiv (\top \wedge (\neg p \vee q)) \equiv (\neg p \vee q)$

IV $(p \wedge (\neg p \vee q)) \equiv ((p \wedge \neg p) \vee (p \wedge q)) \equiv (\bot \vee (p \wedge q)) \equiv (p \wedge q)$

V $(\neg p \wedge (p \vee q)) \equiv ((\neg p \wedge p) \vee (\neg p \wedge q)) \equiv (\bot \vee (\neg p \wedge q)) \equiv (\neg p \wedge q)$

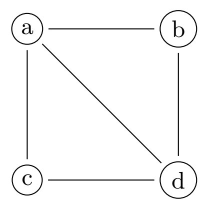
Each of these classes are distinct, as can be seen from the truth table:

p	q	$\neg p$	I	II	III	IV	V
T	T	F	T	T	T	T	F
T	F	F	T	F	F	F	F
F	T	T	T	T	T	F	T
F	F	T	F	T	F	F	F

So there are five equivalence classes.

(10 marks)

4. Consider the following graph:



(a) Suppose you colour each vertex one of three colours chosen uniformly at random. What is the probability you get a valid 3-colouring of the graph?

(b) Suppose you give each edge one of two directions chosen uniformly at random (e.g. alphabetically increasing vs alphabetically decreasing). What is the probability you get a DAG?

Solution:

(a) There are $4^3 = 64$ ways of colouring the four vertices of this graph one of three colours. In terms of valid 3-colourings, we observe that once we have chosen colours for a and b ($3 \times 2 = 6$ possibilities) the colours for c and d are forced. So there are only 6 valid 3-colourings, giving a probability of $\frac{6}{64} = \frac{3}{32}$ that the colouring is valid.

(b) There are $2^5 = 32$ ways of orienting the edges. We could manually count which of these yield DAGs, but a slightly more clever way is as follows. We observe that each permutation of a, b, c, d corresponds to a topological sort of of a DAG for exactly one orientation (so there are at most $4! = 24$ DAGs). There are permutations which correspond to the same orientation, however since only b and c will be "incomparable" in any given orientation we can identify these permutations as those where b and c occur "together" (e.g. *abcd* and *acbd* correspond to the same orientation). So 12 permutations correspond to only 6 orientations (and the remaining 12 permutations give rise to unique orientations) – meaning there are, in total, 18 orientations that are DAGs. So the probability of a DAG is $\frac{18}{64} = \frac{9}{32}$.

(10 marks)