COMP 2711 Discrete Mathematical Tools for Computer Science
2020 Spring Semester – Final Exam (Part 1)

Note 1: For all non-proof questions, showing your steps is not necessary unless required otherwise. However, in case your answer is wrong, showing your steps may earn you some partial credits.
Note 2: You can express your answers using binomial coefficients, factorials, and D_n (derangement number). However, you should not have summation $\sum$ in your final answers. For example, $\binom{10}{3}D_9+4!$ and $1!+2!+3!+4!$ are valid, but $\sum_{i=0}^{n}\binom{n}{i}$ or $\binom{n}{0}+\binom{n}{1}+\cdots+\binom{n}{n}$ is not. The latter has to be simplified to 2^n.

Question 1: (10 pts) Let G be a simple undirected graph with a set of vertices V. Let V_1 and V_2 be subsets of V so that $V_1 \cup V_2 = V$ and $V_1 \cap V_2 = \emptyset$. Let $E(x, y)$ be the predicate representing that there is an edge from x to y. Note that the graph being undirected means that $\forall u \in V\ \forall v \in V(E(u, v) \Leftrightarrow E(v, u))$.

(a) (6 pts) Express each of the following properties in predicate logic. Your can only use V, V_1, V_2, $E(\cdot,\cdot)$, logical and mathematical operators.

(i) "Every vertex in the graph has degree exactly 1".
(ii) "Every edge connects a vertex in V_1 and a vertex in V_2".
(iii) "For every vertex in V_1, there are edges that connect it with all vertices in V_2".

(b) (2 pts) If (a)(ii) is true, is G necessarily a bipartite graph? Your answer should be either "Yes" or "No". Justification is not necessary.

(c) (2 pts) If (a)(iii) is true, is G necessarily a complete bipartite graph? Your answer should be either "Yes" or "No". Justification is not necessary.

Solution: (a) (i) $\forall u \in V(\exists v \in V(E(u, v) \wedge \forall w \in V(E(u, w) \to v = w)))$.
(ii) $(\forall u \in V_1 \forall v \in V_1(\neg E(u, v))) \wedge (\forall u \in V_2 \forall v \in V_2(\neg E(u, v)))$.
(iii) $\forall u \in V_1 \forall v \in V_2(E(u, v))$.

(b) Yes.

(c) No.

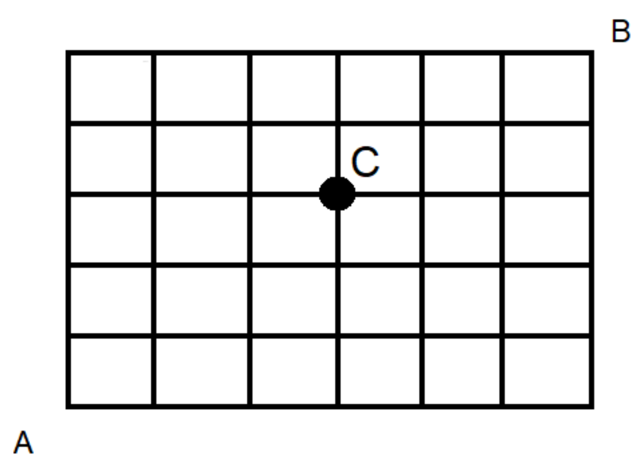
Question 2: (8 pts) Prove that for any $n \in \mathbb{N}$, there exists an n-digit number A_n such that all its digits are either 1 or 2, and $2^n \mid A_n$. [Hint: Use induction.]

Solution: We use induction on n.
Base case: $n = 1$. We can set $A_1 = 2,\ \ 2^1 \mid A_1$.
Induction hypothesis: Assume it is correct for $n = k$, which means that there exists A_k such that $2^k \mid A_k$.

Now consider $n = k+1$: As $2^k \mid A_k$, there are two possibilities for the remainder of A_k dividing 2^{k+1}. $A_k = 2^k \cdot l$, where l is even, which means $A_k \equiv 0 \pmod{2^{k+1}}$, or $A_k = 2^k \cdot (1 + l')$, where l' is even, which means $A_k \equiv 2^k \pmod{2^{k+1}}$

1. $A_k \equiv 0 \pmod{2^{k+1}}$. In this scenario we add the digit 2 to the left side of A_k. So $A_{k+1} = \overline{2A_k}$. We show that $2^{k+1} \mid A_{k+1}$.
   $A_{k+1} = 2 \cdot 10^k + A_k = 2 \cdot 2^k \cdot 5^k + A_k = 2^{k+1} \cdot 5^k + A_k$. And we know that $2^{k+1} \cdot 5^k \equiv 0 \pmod{2^{k+1}}$ and $A_k \equiv 0 \pmod{2^{k+1}}$. So $A_{k+1} \equiv 0 \pmod{2^{k+1}}$.
2. $A_k \equiv 2^k \pmod{2^{k+1}}$. In this scenario we add the digit 1 to the left side of A_k. So $A_{k+1} = \overline{1A_k}$. We show that $2^{k+1} \mid A_{k+1}$.
   $A_{k+1} = 1 \cdot 10^k + A_k = 2^k \cdot 5^k + A_k$. And we know that $A_k \equiv 2^k \pmod{2^{k+1}}$. So $A_{k+1} \equiv 2^k + 2^k \cdot 5^k \equiv 2^k \cdot (1 + 5^k) \pmod{2^{k+1}}$.
   $1 + 5^k$ is an even number so $A_{k+1} \equiv 0 \pmod{2^{k+1}}$.

Question 3: (5 pts) You are staying on point A and you are allowed to walk either up or to the right. Answer these questions:



(a) (2 pts) In how many ways you can go to point B?

(b) (3 pts) In how many ways you can go to point B without passing through point C?

Solution: (a) $\binom{11}{5}$

(b) $\binom{11}{5} - \binom{6}{3} \cdot \binom{5}{3}$

Question 4: (14 pts) How many integers $0 \leq n < 1000$ are there such that

(a) (3 pts) n is divisible by 2, 3 or 5?

(b) (4 pts) $n \equiv 1 \pmod 2$, $n \equiv 2 \pmod 3$, and $n \equiv 3 \pmod 5$?

(c) (7 pts) n is divisible by 9 but it has no digit 9?

Solution: (a) Let $f(k)$ be the number of integers $0 \le n < 1000$ that are divided by k, then $f(k) = 1 + \lfloor 999/k \rfloor$. We have $f(2) = 500$, $f(3) = 334$, $f(5) = 200$, $f(6) = 167$, $f(10) = 100$, $f(15) = 67$ and $f(30) = 34$. By inclusion-exclusion principle, the answer is $f(2)+f(3)+f(5)-f(6)-f(10)-f(15)+f(30) = 734$.

(b) By Chinese remainder theorem, we know that $n \equiv 1 \pmod 2$, $n \equiv 2 \pmod 3$, and $n \equiv 3 \pmod 5$ is equivalent to $n = 23 + 30k$ for an integer k. Since $0 \le n < 1000$, we have $0 \le k \le 32$. Therefore the answer is 33.

(c) Let $n = 100x_1 + 10x_2 + x_3$, i.e. n is represented by 3 digits $x_1\bar{x}_2x_3$. Then $x_i \in \{0, 1, \cdots, 8\}$ for $i = 0, 1, 2$, and $9 | 100x_1 + 10x_2 + x_3$, or equivalently $9 | x_1 + x_2 + x_3$. Since $0 \le x_1 + x_2 + x_3 \le 24$, we have $x_1 + x_2 + x_3 = 0, 9$ or 18.

(1) $x_1 + x_2 + x_3 = 0$. There is only one solution: $x_1 = x_2 = x_3 = 0$.

(2) $x_1 + x_2 + x_3 = 9$. By the combinations with repetition theorem, there are $\binom{9+3-1}{3-1} = 55$ non-negative solutions to this equation. However we need to exclude three solutions $(9, 0, 0)$, $(0, 9, 0)$ and $(0, 0, 9)$. Therefore the number of solutions is 52.

(3) $x_1+x_2+x_3 = 18$. First note that there is no solution when $x_3 = 0$ or 1. Suppose $x_3 = i$ where $i \in \{2, 3, \cdots, 8\}$, the solutions to the equation $x_1+x_2 = 18-i$ are $(j, 18-i-j)$ where $j \in \{10-i, 11-i, \cdots, 8\}$, i.e. there are $i-1$ solutions to this equation. Therefore the number of solutions to $x_1 + x_2 + x_3 = 18$ is $\sum_{i=2}^{8}(i-1) = 28$.

Therefore we conclude that the answer is $1 + 52 + 28 = 81$.

Alternative Solution: We also find the number of solutions to $9 | x_1 + x_2 + x_3$ where $x_i \in \{0, 1, \cdots, 8\}$ for $i = 1, 2, 3$. For any x_1 and x_2, $9 | x_1 + x_2 + x_3$ is equivalent to $x_3 \equiv 9 - x_1 - x_2 \pmod 9$, which has exactly one solution in $\{0, 1, \cdots, 8\}$. Therefore the number of solutions is $9 \times 9 = 81$.

Question 5: (10 pts) Consider the following series of functions defined recursively: $f_0(x) = x^{-1}$; for $n \ge 0$, $f_{n+1}(x) = f_n'(x)$ (recall that $(ax^b)' = abx^{b-1}$).

(a) (5 pts) Find the general formula of $f_n(x)$ and use induction to prove its correctness.

(b) (5 pts) Let $a_n = \log(-f_n(-1))$. Which one of the following functions is the tight asymptotic bound (i.e., Θ) of a_n? Justify your answer.

$$n, n \log \log n, n \log n, n \log^2 n, n^2$$

Solution: (a) $f_n(x) = (-1)^n n! \cdot x^{-(n+1)}$. We can prove this by induction. First $f_0(x) = (-1)^0 \cdot 0! \cdot x^{-1} = x^{-1}$. Suppose for some $k \geq 0$, $f_k(x) = (-1)^k k! \cdot x^{-(k+1)}$, then

$$f_{k+1}(x) = f_k'(x) = (-1)^k k! \cdot [-(k+1)] \cdot x^{-(k+2)} = (-1)^{k+1}(k+1)! \cdot x^{-(k+2)}.$$

(b) $a_n = \log n! = \sum_{i=1}^{n} \log i$. On the one hand, $a_n < \sum_{i=1}^{n} \log n = n \log n$, so $a_n = O(n \log n)$. On the other hand, $a_n > \sum_{i=n/2+1}^{n} \log(n/2) = \frac{n}{2} \log \frac{n}{2}$, so $a_n = \Omega(n \log n)$. Therefore we conclude $a_n = \Theta(n \log n)$, and the tightest asymptotic upper bound is $n \log n$.

Question 6: (6 pts) Read the following algorithms and answer the questions.

Algorithm 1: `MultiplyTwoPolys(`$p(x),\ q(x)$`)`

Input: A polynomial $p(x)$ with degree m and a polynomial $q(x)$ with degree n
Output: The polynomial $r(x) = p(x) \cdot q(x)$
for $i \leftarrow 0$ **to** m **do**
  $a_i \leftarrow$the coefficient of x^i in $p(x)$;
end
for $i \leftarrow 0$ **to** n **do**
  $b_i \leftarrow$the coefficient of x^i in $q(x)$;
end
for $i \leftarrow 0$ **to** $m + n$ **do**
  $c_i \leftarrow 0$;
end
for $i \leftarrow 0$ **to** m **do**
  for $j \leftarrow 0$ **to** n **do**
    $c_{i+j} \leftarrow c_{i+j} + a_i b_j$;
  end
end
$r(x) \leftarrow \sum_{i=0}^{m+n} c_i x^i$;
return $r(x)$;

Algorithm 2: `MultiplyManyPolys(`$p_1(x), p_2(x), \cdots, p_n(x)$`)`

Input: Polynomials $p_1(x), p_2(x), \cdots, p_n(x)$
Output: The polynomial $r(x) = p_1(x)p_2(x)\cdots p_n(x)$
if $n = 1$ **then**
  return $p_1(x)$;
end
$r_1(x) \leftarrow$`MultiplyManyPolys(`$p_1(x), p_2(x), \cdots, p_{n-1}(x)$`)`;
$r(x) \leftarrow$`MultiplyTwoPolys(`$r_1(x),\ p_n(x)$`)`;
return $r(x)$;

(a) (2 pts) Algorithm 1 computes the multiplication of two polynomials with degree m and n respectively. What is its time complexity? Write your answer in the best big-O notation with respect to m and n.

(b) (4 pts) Let $\{p_n(x)\}$ be a sequence of polynomials, all having degree 1. What is the time complexity of computing $p_1(x)p_2(x)\cdots p_n(x)$ using Algorithm 2? Write your answer in the best big-O notation with respect to n. Justify your answer.

Solution:

(a) The time complexity of algorithm 1 is $O(mn)$.

(b) Let C_n be the time complexity of computing $p_1(x)p_2(x)\cdots p_n(x)$ using Algorithm 2, then $C_1 = 0$ and $C_n = C_{n-1} + n - 1$. Therefore $C_n = \sum_{i=1}^{n}(i-1) = O(n^2)$.