Math for CS 2015/2019 Problem Set 6 solutions

https://github.com/spamegg1

May 27, 2022

Contents

1 Problem 1 **1**

2 Problem 2 **2**
2.1 (a) 2
2.2 (b) 3

3 Problem 3 **3**
3.1 (a) 4
3.2 (b) 4
3.3 (c) 4

1 Problem 1

Suppose the RSA modulus $n = pq$ is the product of distinct 200 digit primes p and q. A message $m \in [0..n)$ is called *dangerous* if $\gcd(m, n) = p$ or $\gcd(m, n) = q$, because such an m can be used to factor n and so crack RSA.

Estimate the fraction of messages in $[0..n)$ that are dangerous to the nearest order of magnitude.

Proof. 1. We can list the multiples of p and q below $n = pq$:

$p, 2p, 3p, \ldots, (q-1)p$, and $q, 2q, 3q, \ldots, (p-1)q$.

2. For any number $m = ip$ in the first list we have: $\gcd(m, n) = \gcd(ip, pq) = p$ because $1 \leq i \leq q - 1$ and i is relatively prime to q.

3. For any number $m = iq$ in the second list we have: $\gcd(m, n) = \gcd(iq, pq) = q$ because $1 \leq i \leq p - 1$ and i is relatively prime to p.

4. Now we need to prove that these are all the numbers $m \in [0..n)$ with the property: $\gcd(m, n) = p$ or $\gcd(m, n) = q$.

5. Assume $m \in [0..n)$ has the property: $\gcd(m, n) = p$ or $\gcd(m, n) = q$.

6. Case 1 of (5): $\gcd(m, n) = p$. Then there exists some integer k such that $m = pk$. By definition of gcd, k is relatively prime to q. Since $pk = m < n = pq$, we have $k < q$.

7. Since q is prime, $k < q$ and k is relatively prime to q, all the possibilities for k are $1, 2, 3, \ldots, q - 1$.

8. Case 2 of (5): $\gcd(m, n) = p$. Very similar to steps (6) and (7).

9. (7) and (8) together prove (4). So there are exactly $q-1+p-1$ dangerous messages $m \in [0..n)$.

10. By (9), the fraction of dangerous messages is:

$$\frac{p + q - 2}{pq}$$

11. Since p and q are both 200-digit primes, let's write $p = p' \cdot 10^{200}$ and $q = q' \cdot 10^{200}$ for some rational numbers $0.1 < p', q' < 1$. Then ignoring the -2 in our estimation, we have:

$$\frac{p+q}{pq} = \frac{p' \cdot 10^{200} + q' \cdot 10^{200}}{p' \cdot 10^{200} \cdot q' \cdot 10^{200}} = \frac{10^{200}(p' + q')}{10^{400} \cdot p' \cdot q'} = \frac{1}{10^{200}} \frac{p' + q'}{p' \cdot q'} < \frac{1}{10^{200}} \frac{1+1}{0.1 \cdot 0.1}$$

12. By (11) the fraction is less than $2 \cdot 10^{-198}$. □

2 Problem 2

2.1 (a)

Give an example of a digraph with two vertices $u \neq v$ such that there is a path from u to v and also a path from v to u, but no cycle containing both u and v.

Proof. By definition of walks, paths and cycles on page 321, a walk is allowed to go through the same vertex more than once, but paths and cycles are not.

So consider this digraph: $u \leftrightarrow w \leftrightarrow v$ (here $\leftrightarrow$ means there are two directed edges in both directions).

There is a path from u to v: $u \to w \to v$

There is a path from v to u: $u \leftarrow w \leftarrow v$

But the graph has no cycles containing both u and v. A cycle is a positive length walk that has distinct vertices except for the beginning and end. So the only 4 cycles of this graph are $u \to w \to u$, $w \to u \to w$, and $w \to v \to w$, $v \to w \to v$.

The only way to start and finish at u would be: $u \to w \to v \to w \to v$ but this has to pass through w twice. Similar for v. □

2.2 (b)

Prove that if there is a positive length walk in digraph that starts and ends at node v, then there is a cycle that contains v.

Proof. Assume there is a positive length walk

$$v = v_0\langle v_0 \to v_1\rangle v_1\langle v_1 \to v_2\rangle v_2 \ldots \langle v_{k-1} \to v_k\rangle v_k = v$$

where the beginning and end nodes are both v.

This would be a cycle, if it had all distinct nodes (except v). So we have to prove that we can remove repeated nodes from this walk to obtain a cycle.

Assume that for some $i < j \in \{1, \ldots, k-1\}$ the nodes v_i and v_j are repeated, that is, $v_i = v_j$.

Then we obtain a new positive length walk by removing the redundant segment that comes right after the first occurrence of v_i and ends at v_j. We remove:

$$\langle v_i \to v_{i+1}\rangle v_{i+1}\langle v_{i+1} \to v_{i+2}\rangle v_{i+2} \ldots \langle v_{j-1} \to v_j\rangle v_j$$

We remove these segments for all the repeated nodes v_i, v_j like above, except for the beginning and ending nodes v. The remaining walk is a cycle containing v: it still has positive length, but now without repeated nodes. □

3 Problem 3

Suppose that there are n chickens in a farmyard. Chickens are rather aggressive birds that tend to establish dominance in relationships by pecking; hence the term "pecking order." In particular, for each pair of distinct chickens, either the first pecks the second or the second pecks the first, but not both. We say that chicken u virtually pecks chicken v if either:

Chicken u directly pecks chicken v, or

Chicken u pecks some other chicken w who in turn pecks chicken v.

A chicken that virtually pecks every other chicken is called a king chicken.

We can model this situation with a chicken digraph whose vertices are chickens with an edge from chicken u to chicken v precisely when u pecks v.

In the graph in Figure 1, three of the four chickens are kings. Chicken c is not a king in this example since it does not peck chicken b and it does not peck any chicken that pecks chicken b. Chicken a is a king since it pecks chicken d, who in turn pecks chickens b and c.

In general, a tournament digraph is a digraph with exactly one edge between each pair of distinct vertices.

3.1 (a)

Define a 10-chicken tournament graph with a king chicken that has outdegree 1.

Proof. Name the chickens $c_0, \ldots, c_9$. We will describe a tournament where c_0 is a king with outdegree 1.

By the definition of a tournament ("in particular, for each pair of distinct chickens, either the first pecks the second or the second pecks the first, but not both."), between every pair c_i, c_j of chickens, there has to be exactly one directed edge.

To make sure c_0 is a king, we can use a chain of pecking $c_0 \to c_1 \to c_2 \to \ldots \to c_9$. This way, c_0 virtually pecks every other chicken.

To make sure c_0 has outdegree 1, the only outgoing edge from c_0 is $c_0 \to c_1$. For all the other chickens, they all peck c_0 instead: $c_2 \to c_0, c_3 \to c_0, \ldots, c_9 \to c_0$.

Then the remaining directed edges between the other chicken $c_1, \ldots, c_9$ that need to be filled in can be whatever, doesn't matter as long as we don't change the above setup. □

3.2 (b)

Describe a 5-chicken tournament graph in which every player is a king.

Proof. We can use the same idea as in part (a). Create a closed 5-cycle $c_0 \to c_1 \to c_2 \to \ldots \to c_4 \to c_0$. This way, every chicken virtually pecks every other chicken. So every chicken is a king. The missing edges between pairs of chicken can be however we want, doesn't matter. □

3.3 (c)

Prove

Theorem. *(King Chicken Theorem) The chicken with the largest outdegree in an n-chicken tournament is a king.*

The King Chicken Theorem means that if the player with the most victories is defeated by another player x, then at least he/she defeats some third player that defeats x. In this sense, the player with the most victories has some sort of bragging rights over every other player. Unfortunately, as Figure 1 illustrates, there can be many other players with such bragging rights, even some with fewer victories.

Proof. 1. Assume the size of the tournament is n, and there are n chickens: $c_1, \ldots, c_n$. For a chicken c define $out(c)$ to be the outdegree and $in(c)$ to be the indegree of c.

2. Argue by contradiction and assume that, say chicken c_1 has the largest outdegree $out(c_1)$ among all the chicken, but is not a king.

3. By definition of a king, there exists another chicken, say c_2, such that c_1 does not virtually peck c_2. (This means there is no directed path from c_1 to c_2 in the graph.) Also, by (2) we have $out(c_1) \geq out(c_2)$.

4. Since c_1 does not virtually peck c_2, c_1 does not directly peck c_2 either (there is no edge $c_1 \to c_2$).

5. By definition of a tournament, either c_1 pecks c_2 or vice versa. So by (3) c_2 pecks c_1 (there is an edge $c_2 \to c_1$).

6. Consider the set C_1 of all the chicken who are directly pecked by c_1. So $out(c_1) = |C_1|$.

7. For any chicken $c \in C_1$, c does not directly peck c_2, otherwise c_1 would virtually peck c_2 via the path $c_1 \to c \to c_2$, which would contradict (3).

8. By (7) and again by definition of a tournament, c_2 directly pecks every chicken $c \in C_1$.

9. By (5) and (8) $out(c_2)$ is at least $|C_1| + 1$, so $out(c_2) > |C_1| = out(c_1)$, contradiction to (3).

10. So our initial assumption was false, and c_1 must be a king. □