Matthew Feng October 26, 2018

6.046 Problem Set 6

Collaborators: *James Lin, Katherine Xiao*

Problem 1

(a)

If there are no cycles in G, that means that the number of edges in every connected component of G is one less than the number of nodes in that connected component. That means that there more valid slots available than items we want to insert, and thus it is possible to place the items into the one of the two slots that h_1 and h_2 assign them to.

In particular, for every connected component κ in G, perform DFS from an arbitrary start vertex in κ. Every edge e_i corresponds to some value x_i that we wish to put into our hash table. Every time we traverse down an edge $e_i = (u, v)$, we assign the x_i corresponding to the edge to the vertex that we arrive at (i.e. v, if we started from u). Note that this means the start vertex (slot) we chose will not be assigned a value.

The algorithm is correct because DFS traverses every edge exactly once, meaning that every value we want to insert is assigned to a slot that it hashed to. Moreover, each vertex is visited exactly once, and so each vertex (slot) will never be assigned to two values (i.e. no values will ever be overridden). Since every value is given a slot, and no slot is ever used twice, the algorithm is correct. Furthermore, DFS runs in $O(n)$ time, so the runtime of our algorithm is $O(n)$.

(b)

For a given k-cycle defined by edges $C = \{(m_1, m_2), (m_2, m_3), ..., (m_{k-1}, m_k), (m_k, m_1)\}$ (noting all edges are undirected, and the cycle occurs in that particular order), the probability that an edge $e = (m_a, m_b) \in C$ exists is $2/m^2$, since the edge will exist if either

1. $h_1(x) = m_a$ and $h_2(x) = m_b$, or
2. $h_1(x) = m_b$ and $h_2(x) = m_a$,

both of which have a probability of $1/m^2$ individually of occuring. Since x can be any of the n items, the probability that e exists is $2n/m^2$. We need all k of the edges in C to exist, and each may appear independently, and so the probability that all exist is $(2n/m^2)^k$.

(c)

Since there are at most $\frac{(k-1)!}{2}\binom{m}{k}$ potential ways for a k-cycle to exist in G ($\binom{m}{k}$ ways to choose the vertices, and $\frac{(k-1)!}{2}$ ways to order the cycle), the probability that of any k-cycle appearing in G is upper bounded (union bound) by

$$\frac{(k-1)!}{2}\binom{m}{k}\left(\frac{2n}{m^2}\right)^k \leq \left(\frac{2n}{m}\right)^k.$$

(d)

From (c), we know that the probability of any k-cycle is at most $(\frac{2n}{m})^k$. Then the probability of any cycle is upper bounded (again by union bound) by

$$\sum_{k=1}^{\infty}\left(\frac{2n}{m}\right)^k.$$

If $m = 10n$, then the above sum becomes

$$\sum_{k=1}^{\infty}\left(\frac{1}{5}\right)^k = \frac{1}{4} < \frac{1}{2}.$$

(e)

If we run our algorithm in (a) multiple times (each time generating new hash functions h_1 and h_2), where each will succeed with probability of at least $\frac{1}{2}$, then the chance that Ben fails decreases exponentially. Since each run has at least $\frac{1}{2}$ chance of succeeding, the expected number of runs before a success (including the success) is 2 (by expectation of a geometric random variable). Since the expected number of needed runs is constant, and it takes $O(n)$ time for each call to the algorithm, the expected running time of this hashing procedure is $O(n)$.

Problem 2

(a)

By definition, a zero-sum game requires $F(x, y) + G(x, y) = 0$.

(b)

Beff always has a strategy to ensure that Melon's payoff is $\leq \frac{1}{3}$.

We can split all of Melon's potential strategies into two cases:

Case 1. *The probability that $x < \frac{1}{2}$ is less than or equal to $\frac{1}{3}$.*

In this case, if Beff employs the strategy $y = 1$, then Melon's expected payoff is

$$(\text{some value} \leq \frac{1}{3}) - (\text{some value} \geq \frac{2}{3})$$

which is less than $\frac{1}{3}$.

Case 2. *The probability that $x < \frac{1}{2}$ is greater than $\frac{1}{3}$.*

In this case, if we let $y = a$ such that $\mathbb{P}[x < a] = \frac{1}{3}$, then the maximum expected payoff that Melon can achieve is

$$(\frac{2}{3}) - (\frac{1}{3})$$

where the former term is the positive payoff and the latter is the negative payoff. This payoff sums to $\frac{1}{3}$. Note that $y < \frac{1}{2}$ because in order for $\mathbb{P}[x < \frac{1}{2}]$ to be greater than $\frac{1}{3}$, it must have equaled $\frac{1}{3}$ for some $x < \frac{1}{2}$.

In both cases, which span all of Melon's strategies, Beff has a response that guarantees Melon's payoff is at most $\frac{1}{3}$.

(c)

This time, we want to find ways to choose x so that for any probability distribution on y, Melon has a payoff of at least $\frac{3}{7}$. We again proceed with cases on the strategies that Beff could use.

Case 1. *The probability that $y < 1$ is less than $\frac{3}{7}$.*

This case needs to be further divided into the following two cases:

Case 1a. *The probability that $y < \frac{1}{2}$ is less than or equal to $\frac{1}{7}$.*

In this case, if Melon always selects $x = 0$, then his expected payoff will be

$$(\text{some value} > \frac{4}{7}) - (\text{some value} \leq \frac{1}{7}) + (\text{some value} \geq \frac{2}{7})$$

which is guaranteed to be greater than or equal to $\frac{3}{7}$ (the latter two terms always sum to a non-negative value).

Case 1b. *The probability that $y < \frac{1}{2}$ is greater than or to $\frac{1}{7}$.*

Pick $x = b$ for the b such that $\mathbb{P}[y < b] = \frac{1}{7}$. x is guaranteed to be less than $\frac{1}{2}$ because in order for $\mathbb{P}[y < \frac{1}{2}]$ to exceed $\frac{1}{7}$, $\mathbb{P}[y < \frac{1}{2}]$ must have been equal to $\frac{1}{7}$ (since the cumulative distribution is monotonically increasing). Then, Melon's expected payoff is

$$(\text{some value} > \frac{4}{7}) - (\text{some value} < \frac{2}{7}) + (\frac{1}{7})$$

which is also guaranteed to be greater than or equal to $\frac{3}{7}$. The first term comes from when $y = 1$, the third from the assumption in the case, and the second is the worst case in that the remaining distribution of y is in the -1 "zone".

Case 2. *The probability that $y < 1$ is greater than or equal to $\frac{3}{7}$.*
In this case, if Melon always selects $x = 1$, then his payoff will be

$$(\text{some value} \geq \frac{3}{7}) - 0 \times (\text{some value} \leq \frac{4}{7})$$

which is guaranteed to be greater than $\frac{3}{7}$.

Because the cases cover all possible strategies that Beff could pick, Melon always can respond with a strategy that guarantees him a payoff of at least $\frac{3}{7}$.

(d)

No, since the minimum payoff for Melon if Beff chooses his strategy first does not equal the maximum payoff he can ensure if Beff chooses his strategy second.

(e)

This is not a Nash Equilibrium, because either Melon or Beff can change their strategy (e.g. $x = 0$) to gain a higher payoff.