Randomized Algorithms, Spring 2013
Exercise 5: Solutions

---

1. **Exercise 5.21:** In hashing with *open addressing*, the hash table is implemented as an array and there are no linked list or chaining. Each entry in the array either contains one hashed item or is empty. The hash function defines, for each key k, a *probe sequence* $h(k,0), h(k,1), \ldots$ of table locations. To insert the key k, we first examine the sequence of table locations in the order defined by the key's probe sequence until we find and empty location; then we insert the item at that position. When searching for an item in the hash table, we examine the sequence of table locations in the order defined by the key's probe sequence until either the item is found or we have found an empty location in the sequence. If an empty location is found, this means the item is not present in the table.
An open-address hash table with $2n$ entries is used to store n items. Assume that the table location $h(k,j)$ is uniform over the $2n$ possible table locations and that all $h(k,j)$ are independent.

a) Show that, under these conditions, the probability of an insertion requiring more than k probes is at most 2^{-k}.

Let $i < n$ be the number of items stored in the has table. Since $h(k,j)$ is uniform over $2n$ locations, the probability that a probe finds an empty location is $(2n-i)/(2n) > 1/2$ and the probability that a probe fails is at most $1/2$. An insertion requires more than k probes if the first k probes fail. Since the probes are independent, the probability that this happens is at most $(1/2)^k = 2^{-k}$.

b) Show that, for $i = 1,2,\ldots,n$, the probability that the ith insertion requires more than $2\log n$ probes is at most $1/n^2$.

Based on (a), by setting $k = 2\log n$ (and assuming that the base of the logarithm is 2) we get that the probability is at most

$$2^{-2\log n} = \frac{1}{n^2}.$$

Let the random variable X_i denote the number of probes required by the ith insertion. You have shown in part (b) that $\Pr(X_i > 2\log n) \leq 1/n^2$. Let the random variable $X = \max_{1\leq i\leq n} X_i$ denote the maximum number of probes required by any of the n insertions.

c) Show that $\Pr(X > 2\log n) \leq 1/n$.

Using a union bound and the result from (b) we get

$$\Pr(X > 2\log n) = \Pr\left(\bigcup_i (X_i > 2\log n)\right) \leq \sum_i \Pr(X_i > 2\log n) \leq n \cdot \frac{1}{n^2} = \frac{1}{n}.$$

d) Show that the expected length of the longest probe sequence is $\mathbf{E}[X] = O(\log n)$.

Let $t = \log_2 n$. The idea is to divide the expectation into two separate parts as follows

$$\begin{aligned}
\mathbf{E}[X] &= \sum_{j=1}^{\infty} j \Pr(X = j) \\
&= \sum_{j=1}^{t} j \Pr(X = j) + \sum_{j=t+1}^{\infty} j \Pr(X = j) \\
&\leq \sum_{j=1}^{t} t \Pr(X = j) + \sum_{j=t+1}^{\infty} t \Pr(X = j) + \sum_{j=t+1}^{\infty} (j-t) \Pr(X = j) \\
&= t \sum_{j=1}^{\infty} \Pr(X = j) + \sum_{j=t+1}^{\infty} (j-t) \Pr(X = j) \\
&= t + S,
\end{aligned}$$

where $S = \sum_{j=t+1}^{\infty}(j-t)\Pr(X=j)$. Since $t = \log_2 n$, it remains to show that S is small enough. We have

$$\begin{aligned} S &= \sum_{j=t+1}^{\infty}\sum_{k=t}^{j-1}\Pr(X=j) \\ &= \sum_{k=t}^{\infty}\sum_{j=k+1}^{\infty}\Pr(X=j) \\ &= \sum_{k=t}^{\infty}\Pr(X>k). \end{aligned}$$

Like in (c), we can use a union bound and the result from (a) to get that

$$\Pr(X>k) = \Pr\left(\bigcup_i (X_i > k)\right) \leq \sum_i \Pr(X_i > k) \leq n2^{-k}.$$

Plugging this into above, we obtain

$$S \leq \sum_{k=t}^{\infty} n2^{-k} = n2^{-t}\sum_{j=0}^{\infty}2^{-j} = n2^{-\log_2 n}\cdot 2 = 2.$$

Thus, the expectation of the longest probe sequence is

$$\mathbf{E}[X] \leq t + S \leq \log_2 n + 2 = O(\log n).$$

---

2. **Exercise 5.22:** Bloom filters can be used to estimate set differences. Suppose you have a set X and I have a set Y, both with n elements. For example, the sets might represent 100 our favorite songs. We both create Bloom filters of our sets, using the same number of bits m and the same k hash functions. Determine the expected number of bits where our Bloom filters differ as a functions of m, n, k, and $|X \cap Y|$. Explain how this could be used as a tool to find people with the same taste in music more easily than comparing lists of songs directly.

Let $c = |X \cap Y|$, and let A_X and A_Y be the bit arrays of the Bloom filters of X and Y respectively.

Now $A_X[i] \neq A_Y[i]$ if and only if

1. no element $x \in X \cap Y$ maps to bit i and
2. either
   a) at least one element $x \in X \setminus Y$ maps to the ith bit but no element in $x \in Y \setminus X$ maps to the ith bit or
   b) at least one element $x \in Y \setminus X$ maps to the ith bit but no element in $x \in X \setminus Y$ maps to the ith bit

Denote by $p = 1 - 1/m$ the probability that a single call to a hash function does not map to a fixed bit. Now, the probability that condition 1. holds is p^{ck}. In condition 2.a, the probability that no element in $x \in Y \setminus X$ sets the ith bit is $p^{(n-c)k}$, and the probability that at least one element $x \in X \setminus Y$ sets the ith bit is $1 - p^{(n-c)k}$. We get similar probabilities for condition 2.b. As 2.a and 2.b are distinct, the probability that either happens is $2p^{(n-c)k}(1-p^{(n-c)k})$. Since conditions 1. and 2. are independent, we have that

$$\Pr(A_X[i] \neq A_Y[i]) = p^{ck}\cdot 2p^{(n-c)k}(1-p^{(n-c)k}) = 2p^{nk}(1-p^{(n-c)k}).$$

The above holds for all i. Now, using the linearity of expectations, we get that

$$\mathbf{E}[\text{the number of bits that differ}] = 2mp^{nk}(1-p^{(n-c)k}).$$

We see that the expectation is increasing with respect to c (that is, $|X \cap Y|$). Thus, the larger is the number of same songs on the lists, the less the Bloom filters differ.

3. **Exercise 5.26:** Consider Algorithm 5.2, the modified algorithm for finding Hamiltonian cycles. We have shown that the algorithm can be applied to find a Hamiltonian cycle with high probability in a graph chosen randomly from $G_{n,p}$, when p is known and sufficiently large, by initially placing edges in the edge lists appropriately. Argue that the algorithm can similarly be applied to find a Hamiltonian cycle with high probability on a graph chosen randomly from $G_{n,N}$ when $N = c_1 n \ln n$ for a suitably large constant c_1. Argue also that the modified algorithm can be applied even when p is not known in advance as long as p is at least $c_2 \ln n/n$ for a suitably large constant c_2.

In our proof we utilize the fact that for a graph $G = (V,E)$ and $E' \subseteq E$ any Hamiltonian path in the subgraph $G' = (V,E')$ is also Hamiltonian path in G.

We start by analyzing the Algorithm 5.2 a bit further. Let $p_0 = 40 \ln n/n$ and let $G = (V,E)$ be a random graph from distribution G_{n,p_0}. Let A_G be the event that "the algorithm fails for graph G". According to Corollary 5.17 the probability of this is $P(A_G) = O(1/n)$.

Let $M = 20n \ln n \geq 20(n-1)\ln n = \binom{n}{2} p_0 = \mathbf{E}[|E|]$, where $\mathbf{E}[|E|]$ is the expected number of edges in G. By using Markov's inequality we get that

$$\Pr(|E| \geq 2M) \leq \Pr(|E| \geq 2\mathbf{E}[|E|]) \leq \frac{1}{2}$$

so $\Pr(|E| \leq 2M) \geq 1/2$ and thus we have

$$\Pr(A_G) \geq \Pr(A_G \mid |E| \leq 2M)\Pr(|E| \leq 2M) \geq \frac{1}{2}\Pr(A_G \mid |E| \leq 2M).$$

and therefore

$$\Pr(A_G \mid |E| \leq 2M) = O\left(\frac{1}{n}\right).$$

Thus, the Algorithm 5.2 still works well even if the graphs are drawn from the distribution G_{n,p_0} conditioned on event $|E| \leq 2M$. Now the idea is to modify the input graph from $G_{n,N}$ so that the resulting graph follows this conditional distribution.

Let still $G = (V,E) \sim G_{n,p_0}$. For $0 \leq m \leq 2M$, let

$$r_m = \Pr(|E| = m \mid |E| \leq 2M) = \frac{\Pr(|E| = m)}{\Pr(|E| \leq 2M)} = \frac{\binom{\binom{n}{2}}{m} p_0^m (1-p_0)^{\binom{n}{2}-m}}{\sum_{j=0}^{2M} \binom{\binom{n}{2}}{j} p_0^j (1-p_0)^{\binom{n}{2}-j}},$$

so that $(r_0, \ldots, r_{2M})$ defines a distribution.

Let $G' = (V,E')$ be a random graph from the following distribution:

1. Draw a random graph $\tilde{G} = (V,\tilde{E})$ from $G_{n,2M}$.
2. Choose $j \in \{0,\ldots,2M\}$ according to distribution $(r_0,\ldots,r_{2M})$ (so that $\Pr(j=i) = r_i$).
3. Choose a subset $E' \subseteq \tilde{E}$ of j edges (that is, $|E'| = j$) uniformly at random from $\tilde{E}$.

Now, for any set of edges E_0 we have

$$\Pr(E' = E_0) = \Pr(E = E_0 \mid |E| \leq 2M),$$

that is, E' follows the desired conditional distribution.

Let $N = 2M = 40n \ln n$. Now we can use the following algorithm to find the Hamiltonian cycle:

1. Let $\tilde{G} = (V,\tilde{E}) \sim G_{n,N}$ be the input.
2. Choose $E' \subseteq \tilde{E}$ as above.
3. Run the Algorithm 5.2 for graph $G' = (V,E')$ with $p = p_0$.

Since the distribution of G' is same as the distribution G_{n,p_0} conditioned on $|E| \leq 2M$, the algorithm finds a Hamiltonian cycle with probability $1 - O(1/n)$.

For the remaining part, let $p \geq 8p_0 = 320 \ln n/n$, where the value of p is not known in advance. The algorithm for input from $G_{n,p}$ is as follows:

1. Let $G'' = (V, E'') \sim G_{n,p}$ be the input.
2. If $|E''| < 2M$, then *fail*.
3. Otherwise, choose a random subset $\tilde{E}$ of $2M$ edges from E'' and apply the above algorithm for graph $\tilde{G} = (V, \tilde{E})$.

Since $\mathbf{E}[|E''|] \geq \binom{n}{2} 8p_0 = 160(n-1)\ln n \geq 80n \ln n = 4M$, the probability that the algorithm fails on step 2 is smaller than $1/n$ for large enough n (proof for example by using a Chernoff bound). Thus, the probability that the whole algorithm fails is $O(1/n)$.

---

4. **Exercise 6.4:** Consider the following two-player game. The game begins with k tokens placed at the number 0 on the integer number line spanning $[0, n]$. Each round, one player, called the *chooser*, selects two disjoint and nonempty sets of tokens A and B. (The sets A and B need not cover all the remaining tokens; they only need to be disjoint.) The second player, called the *remover*, takes all the tokens from one of the sets off the board. The tokens from the other set all move up one space on the number line from their current position. The chooser wins if any token ever reaches n. The remover wins if the chooser finishes with one token that has not reached n.

Denote the chooser by C and the remover by R.

a) Give a winning strategy for the chooser when $k \geq 2^n$.

The winning strategy for C is to each round divide all remaining tokens into two sets of equal size (if the number of tokens is odd, then one of the sets gets one token more). This way it does not matter which set R decides to remove, but on jth round at least 2^{n-j} tokens will be moved on position j (since $k \geq 2^n$). Specifically, position n will receive at least $2^0 = 1$ token and thus C will win.

b) Use the probabilistic method to show that there must exist a winning strategy for the remover when $k < 2^n$.

Let's assume that on each round R flips a (fair) coin to choose whether to remove A or B. Let X_i be an indicator variable for the event that the ith token reaches n. This happens if the token is selected n times in either A or B and each time the token avoids being removed. The probability of this event is thus

$$\Pr(X_i = 1) = 2^{-n}.$$

The number of tokens that reach n is then $X = \sum_{i=1}^{k} X_i$ and the expectation of it is therefore

$$\mathbf{E}[X] = \sum_{i=1}^{k} \mathbf{E}[X_i] = k \cdot 2^{-n} < 1.$$

Now, since $\Pr(X \leq \mathbf{E}[X]) > 0$ (Lemma 6.2) and X is a nonnegative integer, there must exist a strategy for R such that $X = 0$ (and thus R wins).

c) Explain how to use the method of conditional expectations to derandomize the winning strategy for the remover when $k < 2^n$.

Let r_i be the choice of R on ith round (so r_i is either A or B). If R always makes such a choice that

$$\mathbf{E}[X \mid r_1, \ldots, r_i] \geq \mathbf{E}[X \mid r_1, \ldots, r_i, r_{i+1}]$$

(note the direction of inequality!) and the game end after m rounds, then

$$\mathbf{E}[X \mid r_1, \ldots, r_m] \leq \mathbf{E}[X] < 1$$

(by induction) and thus R wins. The last inequality follows from part (b). Since in the randomized version r_is are chosen independently and uniformly at random, we have

$$\mathbf{E}[X \mid r_1, \ldots, r_i] = \frac{1}{2}\mathbf{E}[X \mid r_1, \ldots, r_i, r_{i+1} = A] + \frac{1}{2}\mathbf{E}[X \mid r_1, \ldots, r_i, r_{i+1} = B],$$

and thus

$$\mathbf{E}[X \mid r_1, \ldots, r_i] \geq \min\left(\mathbf{E}[X \mid r_1, \ldots, r_i, r_{i+1} = A], \mathbf{E}[X \mid r_1, \ldots, r_i, r_{i+1} = B]\right).$$

Therefore, by choosing r_{i+1} such that $\mathbf{E}[X \mid r_1,\ldots,r_i,r_{i+1}]$ is minimized, R wins. To see how to calculate these two expectations, note that in the middle of the game after $i+1$ rounds the probability that the jth token reaches n is

$$\Pr(X_j = 1 \mid r_1,\ldots,r_i,r_{i+1}) = \begin{cases} 2^{-n+t_{j,i+1}} & \text{if } j \text{ is at position } t_{j,i+1} \text{ after } i+1 \text{ rounds} \\ 0 & \text{if } j \text{ has been removed during the first } i+1 \text{ rounds.} \end{cases}$$

The choices $r_1,\ldots,r_i$ determine, which tokens have been removed before round $i+1$. If T_{i+1} is the set of tokens remaining on round $i+1$, then by the linearity of expectations we get

$$\mathbf{E}[X \mid r_1,\ldots,r_i,r_{i+1} = A] = \sum_{j \in T_{i+1} \setminus A} 2^{-n+t_{j,i+1}} = 2^{-n} \sum_{j \in T_{i+1} \setminus A} 2^{t_{j,i+1}}.$$

Thus, R should remove the set $S \in \{A,B\}$ for which the sum $\sum_{j \in T_{i+1} \setminus S} 2^{t_{j,i+1}}$ is smaller. In particular, if all tokens are at the same position, then the set with larger size should be removed.

---

5. **Exercise 6.10:** A family of subsets $\mathcal{F}$ of $\{1,2,\ldots,n\}$ is called an *antichain* if there is no pair of sets A and B in $\mathcal{F}$ satisfying $A \subset B$.

a) Give an example of $\mathcal{F}$ where $|\mathcal{F}| = \binom{n}{\lfloor n/2 \rfloor}$.

Clearly $\mathcal{F}_k$ that consists of all subsets of size k is an antichain. By selecting all subsets of size $k = \lfloor n/2 \rfloor$, the size of $\mathcal{F}_k$ is as required.

b) Let f_k be the number of sets in $\mathcal{F}$ with size k. Show that

$$\sum_{k=0}^{n} \frac{f_k}{\binom{n}{k}} \leq 1.$$

(*Hint:* Choose a random permutation of the numbers from 1 to n, and let $X_k = 1$ if the first k numbers in your permutation yield a set in $\mathcal{F}$. If $X = \sum_{k=0}^{n} X_k$, what can you say about X?)

Let π be a random permutation of the number from 1 to n (from uniform distribution over all permutations). Denote by $\pi_{1,\ldots,k}$ the set of the first k numbers in π. Let X_k be an indicator variable for the event that $\pi_{1,\ldots,k} \in \mathcal{F}$, and let $X = \sum_{k=0}^{n} X_k$.

Now, since $\mathcal{F}$ is an antichain X_k can be 1 for at most one k. Thus, we have $X \leq 1$ and therefore

$$\mathbf{E}[X] \leq 1.$$

On the other hand, since π uniformly distributed, the distribution of $\pi_{1,\ldots,k}$ is uniform over all $\binom{n}{k}$ subsets of size k. Thus, the probability that $\pi_{1,\ldots,k} \in \mathcal{F}$ happens is $f_k / \binom{n}{k}$ and we get

$$\mathbf{E}[X] = \sum_{k=0}^{n} \mathbf{E}[X_k] = \sum_{k=0}^{n} \frac{f_k}{\binom{n}{k}}.$$

By combining these two observations we get the claim.

c) Argue that $|\mathcal{F}| \leq \binom{n}{\lfloor n/2 \rfloor}$ for any antichain $\mathcal{F}$.

For a fixed n a binomial coefficient $\binom{n}{k}$ is maximized when $k = \lfloor n/2 \rfloor$. Thus, by using the result from (b) we get that

$$\frac{1}{\binom{n}{\lfloor n/2 \rfloor}} \sum_{k=0}^{n} f_k \leq \sum_{k=0}^{n} \frac{f_k}{\binom{n}{k}} \leq 1$$

and therefore $|\mathcal{F}| = \sum_{k=0}^{n} f_k \leq \binom{n}{\lfloor n/2 \rfloor}$.

This result is knows as *Sperner's theorem*, named after a German mathematician Emanuel Sperner. The result of (b) is known as *Lubell–Yamamoto–Meshalkin inequality* (or *LYM inequality*)