Homework 11

Problem 1. Prove that the function family

$$\mathcal{H} = \left\{ h_{a,b} \mid h_{a,b}(x) = a \cdot x + b, a \in \{0,1\}^k, b \in \{0,1\} \right\}$$

is a pairwise independent hash function family for range $R = \{0,1\}$ and domain $U = \{0,1\}^k$.

Solution. We need to prove that $\forall x_1 \neq x_2$ and $\forall y_1, y_2$, we have $\Pr_{h\in\mathcal{H}}(h(x_1) = y_1 \wedge h(x_2) = y_2) = \frac{1}{|R|^2} = \frac{1}{4}$.

Note that:

$$\begin{aligned}
&\Pr_{h\in\mathcal{H}}(h(x_1) = y_1 \wedge h(x_2) = y_2) \\
=&\Pr_{h\in\mathcal{H}}(a \cdot x_1 + b = y_1 \wedge a \cdot x_2 + b = y_2) \\
=&\Pr_{a,b}(a \cdot (x_1 \oplus x_2) = y_1 \oplus y_2 \wedge a \cdot x_2 + b = y_2) \\
=&\sum_i \Pr_b(a \cdot (x_1 \oplus x_2) = y_1 \oplus y_2 \wedge b = a \cdot x_2 + y_2 | a = a_i) \Pr(a_i) \\
=&\sum_i 1_{a_i \cdot (x_1 \oplus x_2) = y_1 \oplus y_2} \times \frac{1}{2} \Pr(a_i) \\
=&\Pr_a(a \cdot (x_1 \oplus x_2) = y_1 \oplus y_2) \times \frac{1}{2}
\end{aligned}$$

The second last equation holds because when $a_i \cdot (x_1 \oplus x_2) = y_1 \oplus y_2$ the event happens when b happens to be the number $a_i \cdot x_2 + y_2$ which is of probability 1/2. In the other case the event can never happen (probability 0).

Since $x_1 \neq x_2$, we have $x_1 \oplus x_2 \neq 0$. If we can prove that $\Pr_a(a \cdot x = y) = \frac{1}{2}$ for any $x \neq 0$, the proof will be done. Say the i-th bit of x is none-zero, $x^i \neq 0$, then we can devide the sample space of a (denote as $\mathcal{A}$) into two parts: $\mathcal{A}_0$ with $a^i = 0$ and $\mathcal{A}_1$ with $a^i = 1$. There is a natural bijection between the two parts by flipping the bit a^i, say $a, \tilde{a}$ only differ in the i-th bit. Then we have $a \cdot x = a^i x^i + \sum_{j \neq i} a^j x^j \neq \tilde{a}^i x^i + \sum_{j \neq i} a^j x^j = \tilde{a} \cdot x$. This indicates that precisely half of the a in $\mathcal{A}$ will make $a \cdot x = 0$ and the other half will make it 1, so $\forall y \in \{0,1\}$, $\Pr_a(a \cdot x = y) = \frac{1}{2}$.

Therefore, by taking $x = x_1 \oplus x_2$ and $y = y_1 \oplus y_2$ we can obtain $\Pr_{h\in\mathcal{H}}(h(x_1) = y_1 \wedge h(x_2) = y_2) = \frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$.

Problem 2.

(a) Consider a random walk $X_0, X_1, X_2, \ldots$ on a chain of $n+1$ vertices $0, 1, \ldots, n$ with the following transition probabilities

$$\Pr(X_t = k | X_{t-1} = j) = \begin{cases} \frac{1}{2} & \text{if } j \in [1, n-1] \text{ and } k = j \pm 1, \\ 1 & \text{if } j = 0, k = 1 \text{ or } j = n, k = n, \\ 0 & \text{otherwise.} \end{cases}$$

Let T_i be the expected number of steps the walk takes to arrive at the end vertex n starting with $X_0 = i$. Prove that $T_i \le n^2$ for all $i \in [0, n]$.

(b) Consider the following randomized algorithm for 2-SAT problems of n variables.

```
Choose an arbitrary initial assignment.
for t = 1, 2, ..., 2n^2 do
    if the current assignment is satisfying then
        Accept immediately.
    else
        Choose an arbitrary clause not satisfied.
        Sample one of the two literals uniformly at random.
        Flip the value of the variable in the sampled literal.
    end if
end for
Reject if haven't accepted.
```

Use Markov inequality to show that the algorithm will find a satisfying solution with probability at least $\frac{1}{2}$ given a yes-instance as input.

Solution. (a) Denote a walk from X_i to X_n as w, and the number of steps it takes as $|w|$. Use w_k to denote the k-th vertex in the walk (starting from $w_0 = X_i$(starting vertex)). Then for $1 \le i \le n-1$:

$$\begin{aligned}
T_i &= \sum_{w | w_0 = X_i} |w| \Pr(w) \\
&= \sum_{w | w_0 = X_i} |w| \sum_{X_j} \Pr(w | w_1 = X_j) \Pr(w_1 = X_j) \\
&= \sum_{w | w_0 = X_i} |w| \Big(\Pr(w | w_1 = X_{i+1}) \times \frac{1}{2} + \Pr(w | w_1 = X_{i-1}) \times \frac{1}{2} \Big) \\
&= \frac{1}{2} \sum_{w' | w'_0 = X_{i+1}} (|w'| + 1) \Pr(w') + \frac{1}{2} \sum_{w' | w'_0 = X_{i-1}} (|w'| + 1) \Pr(w') \\
&= \frac{1}{2} \sum_{w' | w'_0 = X_{i+1}} |w'| \Pr(w') + \frac{1}{2} \sum_{w' | w'_0 = X_{i-1}} |w'| \Pr(w') + 1 \\
&= \frac{1}{2} T_{i+1} + \frac{1}{2} T_{i-1} + 1.
\end{aligned}$$

By taking $w = w_0 w'$.

Specifically, we have $T_0 = T_1 + 1$ and $T_n = 0$. By adding the above equations together we have:

$$\begin{aligned}\sum_{i=0}^{n} T_i &= T_1 + 1 + \sum_{i=1}^{n-1}\Big(\frac{1}{2}T_{i+1} + \frac{1}{2}T_{i-1} + 1\Big)\\ &= T_1 + \frac{1}{2}\sum_{i=2}^{n} T_i + \frac{1}{2}\sum_{i=0}^{n-2} T_i + n\\ &= \frac{1}{2}(T_0 + T_1 + T_{n-1} + T_n) + \sum_{i=1}^{n-2} T_i + n.\end{aligned}$$

So we have:

$$\begin{aligned}T_0 + T_{n-1} + T_n &= \frac{1}{2}(T_0 + T_1 + T_{n-1} + T_n) + n\\ \frac{1}{2}T_0 - \frac{1}{2}T_1 + \frac{1}{2}(T_{n-1} + T_n) &= n\\ \frac{1}{2} + \frac{1}{2}T_{n-1} &= n\\ T_{n-1} &= 2n - 1.\end{aligned}$$

By using $T_{i-1} = 2T_i - T_{i+1} - 2$ for $1 \le i \le n-1$ we can inductively prove the rest $T_i (i = 0, 1, 2, ..., n-1)$ are in following form:

$$T_{n-k} = 2kn - k^2$$

by verifying the base case $T_{n-1} = 2n - 1$ and the inductive step:

$$\begin{aligned}T_{n-k-1} &= 2T_{n-k} - T_{n-k+1} - 2\\ &= 2(2kn - k^2) - 2(k-1)n + (k-1)^2 - 2\\ &= (4k - 2k + 2)n - (2k^2 - k^2 + 2k - 1 + 2)\\ &= 2(k+1)n - (k+1)^2.\end{aligned}$$

Thus, by $T_{n-k} - n^2 = -(n-k)^2 \le 0$ we know that $T_{n-k} \le n^2$ for all k, which implies $T_i \le n^2$ for all $i \in [0, n]$.

(b) Consider the input is satisfyable CNF $\phi(x^1, ..., x^n)$. Say a satisifying assignment is $x_*^1, x_*^2, ..., x_*^n$. For any assignment $x^1, ..., x^n$, we define the "distance" of the assignment to the answer is $d(x) = \sum_i (x^i \oplus x_*^i)$. Then $x = x_* \iff d(x) = 0, 0 \le d(x) \le n$.

When we conduct the algorithm, say the initial assignment is x_0 and $d(x_0) = m$. Then after each iteration step, say the current assignment is x_k, we randomly flip a bit in an unsatisfied clause to get x_{k+1}. Say we chose $C = l^i \wedge l^j$, and flipped x_k^i or x_k^j. Since C is unsatisfied we know that there is $x_k^i \neq x_*^i$ or $x_k^j \neq x_*^j$. Since at least one of the two variable isn't the same with x_*, we have at least 1/2 chance to filp a variable that isn't same with x_*, and at most 1/2 chance to flip one that is. In the former case, after flipping x_{k+1} will have one less "wrong" variable, and thus $d(x_{k+1}) = d(x_k) - 1$. In the latter case, $d(x_{k+1}) = d(x_k) + 1$. Specifically, when $d(x_k) = n$, which means all variables are wrong, flipping any variable will make $d(x_{k+1}) = 1$.

Use vertex X_k to denote all assignments x with $d(x) = n - k$. By (a), for any initial state, in the worst case which we have only 1/2 probability to decrease the distance by each flip, the expectation of flips to reach $d(x) = 0$ (Say it's T) is at most n^2.

Since the algorithm will repeat for $2n^2$ steps, the success rate of the algorithm is $\Pr(T \leq 2n^2)$. By Markov's inequality, we know that:

$$\Pr(T \geq 2n^2) \leq \frac{\mathbb{E}(T)}{2n^2} \leq \frac{n^2}{2n^2} = \frac{1}{2}.$$

Therefore, the algorithm will find a satisfying solution with probability at least $\frac{1}{2}$ given a yes-instance as input.