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.

$$\begin{aligned}\Pr[h_{a,b}(x_1) = y_1 \wedge h_{a,b}(x_2) = y_2] &= \Pr_{a,b}[a \cdot x_1 + b = y_1 \wedge a \cdot x_2 + b = y_2] \\ &= \Pr_{a,b}[a \cdot (x_1 \oplus x_2) = y_1 \oplus y_2 \wedge b = y_1 - a \cdot x_1]\end{aligned}$$

Since for any fixed a,$\Pr[b = y_1 - a \cdot x_1] = 1/2$, thus $\Pr[b = y_1 - a \cdot x_1 \mid a \cdot (x_1 \oplus x_2) = y_1 \oplus y_2] = 1/2$. For any $x_1 \neq x_2$, they must be different on at least one bit, say the jth bit. By the following decomposition,

$$a \cdot (x_1 \oplus x_2) = a^{(j)}(x_1^{(j)} \oplus x_2^{(j)}) + \sum_{i \neq j} a^{(i)}(x_1^{(i)} \oplus x_2^{(i)})$$

. Since for fixed any fixed $a^{(i)}, (i \neq j)$, $\Pr[a^{(j)}(x_1^{(j)} \oplus x_2^{(j)}) + \sum_{i \neq j} a^{(i)}(x_1^{(i)} \oplus x_2^{(i)}) = 0] = 1/2$, thus we have that $\Pr_a[a \cdot (x_1 \oplus x_2) = y_1 \oplus y_2] = 1/2$, obtaining

$$\begin{aligned}&\Pr_{a,b}[a \cdot (x_1 \oplus x_2) = y_1 \oplus y_2 \wedge b = y_1 - a \cdot x_1] \\ =&\Pr[b = y_1 - a \cdot x_1 \mid a \cdot (x_1 \oplus x_2) = y_1 \oplus y_2] \Pr_a[a \cdot (x_1 \oplus x_2) = y_1 \oplus y_2] \\ =&\frac{1}{4} = \frac{1}{|R|^2}.\end{aligned}$$

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 $T_i \leq n^2$ 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 $t = 1, 2, \ldots, 2n^2$ **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) We have the following recursion formula for T_i:

$$\begin{cases} T_0 = T_1 + 1, \\ T_i = \frac{1}{2}T_{i-1} + \frac{1}{2}T_{i+1} + 1, i \in [1, n-1], \\ T_n = 0. \end{cases}$$

Solving the recursion, we can obtain that $T_i = n^2 - i^2 \leq n^2$.(Observe that $T_i - T_{i-1} = T_{i+1} - T_i + 2$).

(b) If the 2SAT formula is satisfiable, it has at least one satisfying assignment w. We denote $Y_t = k$ for the assignment w' at time t have exactly k terms that are the same with w. Note that for one unsatisfying clause in the 2SAT formula, it has two possible cases: two terms are contradictory with w, or one term is contradictory with w. The first case flipping either term will set $Y_{t+1} = k + 1$, and for the second case it will set $Y_{t+1} = k + 1$ or $Y_{t-1} = k - 1$ with the same probability $1/2$. Thus in general, we have that $\Pr[Y_{t+1} = k + 1 \mid Y_t = k] \geq 1/2$. Intuitively, this means our walk is biased towards the vertex n, thus the expected time $\tilde{T}_i$ of the walk to n satisfies $\tilde{T}_i \leq T_i \leq n^2$, Thus applying markov inequality

$$\Pr[\text{walk takes more than } 2n^2 \text{ steps}] \leq \frac{\tilde{T}_i}{2n^2} \leq 1/2,$$

implying our result.

You are not required to master the following proof. Now we formally prove that $\tilde{T}_i \leq T_i$. The major difficulty is that at each step, the coin toss of the walk might not be independent. Instead of tossing a coin, we view the process as uniformly sampling a random number s_t in $[0, 1]$, and we set a threshold p_t for each step of the walk (p_t can be a function of the walk history, initial state...). If $s_t \le p_t$, we set $Y_{t+1} = Y_t + 1$, else we set $Y_{t+1} = Y_t - 1$. Note that for X_t, we set $p_t = 1/2$, while for Y_t, we set $p_t \ge 1/2$ in general. Thus it is easy to see that for the same random string $(s_1, \ldots, s_t)$, we will obtain $X_t \le Y_t$. By this observation we can see that $\tilde{T}_i \le T_i$, since if at time t, we have $X_t = n$, then for the same random string, we will also have $Y_t = n$.