

Homework 8 Solutions

Note: These solutions are not necessarily model answers. Rather, they are designed to be tutorial in nature, and sometimes contain a little more explanation than an ideal solution. Also, bear in mind that there may be more than one correct solution. The maximum total number of points available is 34.

1. (a) It is clear that X, Y are pairwise independent. To see that X, Z are pairwise independent, note that for any $x, z \in \{0, 1, \dots, n\}$: 4pts

$$\Pr[X = x \wedge Z = z] = \Pr[X = x \wedge Y = z - x] = \Pr[X = x] \cdot \Pr[Y = z - x] = \frac{1}{(n+1)^2}.$$

A similar argument implies Y, Z are pairwise independent. Finally, the value of Z is completely determined given the values of X, Y , so X, Y and Z are clearly not independent.

- (b) Let X_1, \dots, X_k be k numbers that are chosen independently and uniformly at random from $\{0, 1, \dots, n\}$, $2pts$ and Z their sum. These $k+1$ r.v.'s are k -wise independent but obviously not independent.

2. (a) The running time is $O(n)$. For each of the two sets S_1, S_2 , we run n hashing operations and increment counters n times. Finally, we need to make n comparisons. 2pts

- (b) If S_1 and S_2 are identical, then clearly the i th counters for both tables will match for all i , regardless of the choice of hash function. 1pt

- (c) Suppose S_1 and S_2 are not identical, and furthermore S_1, S_2 are disjoint. Fix some $x \in S_1$, and note from our assumptions that $x \notin S_2$. For each $y \in S_2, y \neq x$, we have by the property of universal hash functions that $\Pr_h[h(x) = h(y)] \leq 1/cn$. Taking a union bound over all $y \in S_2$, we have $\Pr_h[\exists y \in S_2 : h(x) = h(y)] \leq 1/c$. Hence, with probability $1 - 1/c$ over h , we have that $h(y) \neq h(x)$ for all $y \in S_2$. In that case, the $h(x)$ th counter for S_2 is 0, whereas that for S_1 is at least 1, so the algorithm outputs “no”. 6pts

3. (a) It is clear that the new algorithm still outputs “no” with probability 1 on input $x \notin L$. For input $x \in L$, let $Y = \sum_{i=1}^t Y_i$, where Y_i is the indicator r.v. for the event that $\mathcal{A}(x, r_i)$ outputs “yes”. Then, $E[Y_i] \leq 1/2$ and $\text{Var}[Y_i] \leq 1/4$. Hence, $E[Y] \leq s/2$ and (since the Y_i are independent) $\text{Var}[Y] = \sum_{i=1}^t \text{Var}[Y_i] \leq s/4$. The probability that the new algorithm outputs “no” is given by $\Pr[Y = 0]$. Applying Chebyshev’s inequality, we have 6pts

$$\Pr[Y = 0] \leq \Pr[|Y - E[Y]| \geq s/2] \leq \frac{\text{Var}[Y]}{(s/2)^2} = 1/s.$$

This yields the required bound on the error probability.

- (b) As discussed in class and in [MU, Chap. 13], we only need $O(t)$ random bits to sample $s = 1/\delta$ pairwise independent uniform random strings in $\{0, 1\}^t$; the rest of the algorithm is deterministic. 2pts

- (c) We need to run \mathcal{A} $s = 1/\delta$ times in the new scheme, versus $O(\log(\delta^{-1}))$ times in the standard approach. Hence the running times are $O(t/\delta)$ and $O(t \log \delta^{-1})$ respectively. Note, therefore, that the new scheme uses significantly fewer random bits, but at the expense of an increased running time. 2pts

4. (b) The maximum load observed should be in the range 8 to 10. Note that this is a little larger than the value $\frac{\ln n}{\ln \ln n}$, which for $n = 10^6$ is approximately 5. This reflects the effect of lower order terms even for this quite large value of n . 3pts
- (c) With two choices, the maximum load drops dramatically to about 4, with a very small variance. [In fact, it has been proved that the maximum load is asymptotically about $\frac{\ln \ln n}{\ln 2}$ if the balls are allowed to make $d \geq 2$ choices. Note that, as $n \rightarrow \infty$, this is *exponentially* smaller than for the single-choice scheme (because we get to take another log).] 3pts
- (d) Three or four choices (or even a lot more) do not give a significant improvement over the two-choice case. [This is in line with the fact quoted in part (c) above, since $\frac{\ln \ln n}{\ln 2}$ and $\frac{\ln \ln n}{\ln 3}$ only differ by a small constant factor.] 3pts