CPS 430/590.06 **Prof. John Reif**
Design and Analysis of Algorithms **Fall 2013**
Homework 3 **Matt Dickenson**

Question 1

I. Yes, this hash function is universal. Following the proof of theorem 11.5 in CLRS, we can consider two pairs of input $(r_1, r_2), (s_1, s_2)$:

$$\begin{aligned} r &= (r_1 x_1 + r_2 x_2) \mathrm{mod} m \\ s &= (s_1 x_1 + s_2 x_2) \mathrm{mod} m \\ r - s &= x_1(r_1 - s_1) + x_2(r_2 - s_2) \mathrm{mod} m \end{aligned}$$

To avoid collisions, $r-s$ must be nonzero modulo m which means that the following inequality must be satisfied:

$$\begin{aligned} x_1(r_1 - s_1) &\neq x_2(r_2 - s_2) \\ \frac{r_1 - s_1}{x_2} &\neq \frac{r_2 - s_2}{x_1} \end{aligned}$$

As long as r and s are unique inputs and $x_1 \neq x_2$, the probability of collision is sufficiently low that we can regard H as universal.

This method requires $\lceil \log_2 \mathbf{x_1} + \log_2 \mathbf{x_2} \rceil$ random **bits**.

II. Yes, this method is universal. The reasoning is similar to the above. Basically when m is a power of 2, we are keeping only high-order bits of the input (due to the modulo operation). To maintain uniqueness we then require randomness for the lower-order bits. If we suppose that the bit-length of our inputs (a_1, a_2) are k, then **we require** $\log(\mathbf{m} - \mathbf{k})$ **random bits**.

III. No, this method is not universal. Because the choice of functions is randomized, and the range of each function is less than its domain, collisions are difficult to avoid. This method would require $\log m$ random bits for the choice of function.

Question 2

1. To show that selecting u as the the median minimizes $\sum_i |x_i - u| = \sigma$, consider a simple example of $x = 1, 2, 3, 4, 5$. With $u = 3 = \mathrm{med}(x)$, $\sigma = 2 + 1 + 0 + 1 + 2 = 6$. If we shift to $u = 2$ or $u = 4$, $\sigma = 7$.

More generally, if we think of x as a sorted array and $u = x_m$ as the median, we can see that m minimizes the distance from 1 and n, and thus x_m minimizes σ . Recursively, for any $j \in \{1 \ldots m-1\}$, $u = x_m$ also minimizes the distance from $1 + j$ and $n - j$. Thus setting u to the median x_m minimizes the sum of absolute differences.

2. To show that $u = \text{mean}(x) = \frac{\sum_i x_i}{n} = \bar{x}$ minimizes $\sum_i (x_i - u)^2 = \sigma$, we can set the first derivative to zero and verify that the second derivative is positive:

$$\begin{aligned}
0 &= \frac{\partial}{\partial u}\sigma_i(x_i - u)^2 \\
&= \frac{\partial}{\partial u}(n * \bar{x} - n * u)^2 \\
&= 2n^2(u - \bar{x}) \\
\bar{x} &= u \\
0 &< \frac{\partial}{\partial u^2}\sigma_i(x_i - u)^2 \\
&< \frac{\partial}{\partial u}2n^2(u - \bar{x} \\
&< 2n^2
\end{aligned}$$

For any $n > 0$, $0 < 2n^2$, which verifies that the solution $u = \bar{x}$ minimizes $\sum_i (x_i - u)^2 = \sigma$.

Question 3

Let the binary counter be stored in an bit-array A, with indices numbered from the right so that $A[0]$ is the smallest bit. Let the (unspecified) length be k. For the increment operation, $A[0]$ will flip every time. The i^{th} bit flips iff the $(i-1)^{st}$ bit was 1 before the increment operation was called. For any $i \geq k$ the i^{th} bit cannot flip, and for any $i < k$ the increment operation may require a flip that costs 1 unit. Thus for n increment operations the total number of flips is:

$$\begin{aligned}
\sum_{i=0}^{k-1} \lfloor \frac{n}{2^i} \rfloor &< n \times \sum_{i=0}^{\infty} \frac{1}{2^i} &(1)\\
&< n \times 2 &(2)
\end{aligned}$$

so that the amortized cost of incrementing is $O(n)/n = O(1)$.

Similarly, reset requires at most k bit-flips, and because k is constant reset is $O(1)$.

Because the increment and reset operations are both $O(1)$, a sequence of n operations (which can be either increment, reset, or a combination thereof) will require $O(n)$ time.