Advanced Algorithms - Problem Sheet 1

Dom Hutchinson

February 7, 2020

Question - 1.

A hash function family, $H = \{h1, h_2, \dots\}$ is weakly-universal iff for randomly and uniformly chosen $h \in H$, we have $\mathbb{P}[h(x) = h(y)] \leq 1/m$ for $x, y \in S$ with $x \neq y$.
Consider the following hash function families. For each one, prove that it is weakly universal or given a counter-example.

Question 1 a)
Let p be a prime number and $m \in \mathbb{N}$ with $p \geq m$.
Consider the hash function family where you pick at random $a \in [1, p-1]$ and define $h_a : [0, p-1] \to [0, m-1]$ as $h_a(x) = (ax \mod p) \mod m$.

Answer 1 a)
Consider the case where $m = 3$ & $p = 5$.
We have that $a \in [1, 4]$, $h_a : [0, 4] \to [0, 2]$ and $h_a(x) = (ax \mod 5) \mod 3$.
By evaluating every x at every $h_a(\cdot)$ we get

$a\backslash x$	0	1	2	3	4
$h_1(x)$	0	1	2	0	1
$h_2(x)$	0	2	1	1	0
$h_3(x)$	0	0	1	1	2
$h_4(x)$	0	1	0	2	1

By considering this table we see that $\mathbb{P}(h(1) = h(4)) = \frac{2}{4} = \frac{1}{2} > \frac{1}{3} = \frac{1}{m}$.
Thus this <u>is not</u> a *Weakly Universal Hashing Family.*

Question 1 b)
Let p be a prime number and $m \in \mathbb{N}$ with $p \geq m$.
Consider the hash function family where you pick at random $b \in [0, p-1]$ and define $h_b : [0, p-1] \to [0, m-1]$ as $h_b(x) = (x + b \mod p) \mod m$.

Answer 1 b)
Consider the case where $m = 3$ & $p = 5$.
We have that $b \in [0, 4]$, $h_b : [0, 4] \to [0, 2]$ with $h_b(x) = (x + b \mod 5) \mod 3$.
By evaulating every x at every $h_b(\cdot)$ we get

$b\backslash x$	0	1	2	3	4
$h_0(x)$	0	1	2	0	1
$h_1(x)$	1	2	0	1	0
$h_2(x)$	2	0	1	0	1
$h_3(x)$	0	1	0	1	2
$h_4(x)$	1	0	1	2	0

By considering this table we see that $\mathbb{P}(h(0)=h(3))=\frac{2}{5}>\frac{1}{3}=\frac{1}{m}$.
Thus this <u>is not</u> a *Weakly Universal Hashing Family.*

Question 1 c)
Let p be a prime number and $m\in\mathbb{N}$ with $p\geq m$.
Consider the hash function family where you pick at random $a\in[1,p-1]$ & $b\in[0,m-1]$ and define $h_{a,b}:[0,p-1]\to[0,m-1]$ as $h_{a,b}(x)=((ax+b)\mod p)\mod m$.

Answer 1 c)
Let $x,y\in[0,p-1]$ with $x\neq y$.
Define $c:=ax+b$ & $d:=ay+b$.
Note that in order for $(c\mod p)\mod m\not\equiv(d\mod p)\mod m$ we require $c\mod p\not\equiv d\mod p$, this shall be proved first (by contradiction).

Suppose $c\mod p\equiv d\mod p$.
Then $\exists\ e\in\mathbb{Z}$ st $c=d+ep$.
Further

$$\begin{aligned}\implies\quad ax+b&=ay+ep\\ \implies\quad ax&=ay+ep\\ \implies\quad x&=y+\tfrac{e}{a}p\end{aligned}$$

We have a contradiction here. Either $x>p$ (out of bounds), or $x=y$ (contradiction of definitions).
Thus $c\mod p\not\equiv d\mod p$.

Define $f:=c\mod p$ & $g:=d\mod p$.
We have that $f,g\in[0,m-1]$.
If we fix the value of f and allow g to take any of the m values then $\mathbb{P}(f=g)\leq\frac{1}{m}$.
Thus $\mathbb{P}(h(x)=h(y))\leq\frac{1}{m}$ for $x\neq y$.
Thus this <u>is</u> a *Weakly Universal Hashing Family.* □

Question - 2.

Consider a small variant of cuckoo hashing where we use two tables T_1 & T_2 of the same size and hash functions h_1 & h_2.
When inserting a new key x, we first try to put x at position $h_1(x)$ in T_1. If this leads to a collision, then the previously stored key y is moved to position $h_2(y)$ in T_2. If this leasds to another collision, then the next key is again inserted at the appropriate position in T_1, and so on. In some cases, this procedure continues forever *i.e.* the same configuration appears after some steps of moving hte keys around to dissolve collisions.

Question 2 a)
Consider two tables of size 5 each and tow has functions $h_1(k)=k\mod 5$ and $h_2(k)=\left\lfloor\frac{k}{5}\right\rfloor\mod 5$. Insert the keys $\{27,2,32\}$ in this order into initially empty hash tables, and show the result.

Answer 2 b)

1) Insert 27 to $h_1(27)=27\mod 5=2$.
   There are no collisions.

	0	1	2	3	4
T_1	-	-	27	-	-
T_2	-	-	-	-	-

2) Insert 2 to $h_1(2) = 2 \mod 5 = 2$.
Collision with 27.
Move 27 to $h_2(27) = \lfloor \frac{27}{5} \rfloor \mod 5 = 5 \mod 5 = 0$.

	0	1	2	3	4
T_1	-	-	2	-	-
T_2	27	-	-	-	-

3) Insert 32 to $h_1(32) = 32 \mod 5 = 2$.
Collision with 2.
Move 2 to $h_2(2) = \lfloor \frac{2}{5} \rfloor \mod 5 = 0 \mod 5 = 0$.
Collision with 27.
Move 27 to $h_1(27) = 2$.
Collision with 32. Move 32 to $h_2(32) = \lfloor \frac{32}{5} \rfloor \mod 5 = 6 \mod 5 = 1$.

	0	1	2	3	4
T_1	-	-	27	-	-
T_2	2	32	-	-	-

Question 2 c)
Find another key such that its insertion leads to an infinite sequence of key displacements.

Answer 2 c)
Inserting a value x where $h_1(x) = 2$ and $h_2(x) = 0$ will create a value as we will have three values $\{x, 27, 2\}$ trying to fill two places.
The value <u>52</u> would be valid.
If we try inserting 52 we find the following occurs

Insert 52 to $h_1(52) = 52 \mod 5 = 2$.
Collision with 27.
Move 27 to $h_2(27) = 0$.
Collision with 2.
Move 2 to $h_1(2) = 2$.
Collision with 52.
Move 52 to $h_2(52) = \lfloor \frac{52}{5} \rfloor \mod 5 = 10 \mod 5 = 0$.
Collision with 27.
Move 27 to $h_1(27) = 2$.
Collision with 2.
Move 2 to $h_2(2) = 0$.
Collision with 52.
Move 52 to $h_1(52) = 2$.
$\vdots$

Question - 3.

In order to use cuckoo hashing under an unbounded number of key insertions, we cannot have a hash table of fixed size. The size of the hash table has to scale with the number of keys inserted. Suppose that we never delete a key that has been inserted. Consider the following approach with Cuckoo hashing. When the current hash table fills up to its capacity, a new hash table of doubled size is created. All keys are then rehashed to teh new table. Argue that the average time it takes to resize and rebuild the has table, if spread out over all insertions is constant in expectation. That is, the expected amortised cost of rebuilding is constant.

Answer 3