CPTS 515 HW5

Yang Zhang 11529139

1. $H=\{h_i : 1 \le i \le 8\}$ is not universal for hashing arrays with 8 bits. The reason is that each bit only has 2 possible values (either 1 or 0) so no matter which hash function selected from H the $Prob(1) = Prob(0) = 0.5$ Therefore, the probability of collision is $0.5 \times 0.5 = \frac{1}{4}$ which is bigger than $\frac{1}{m}$ $(\frac{1}{8})$, so that H is not universal.

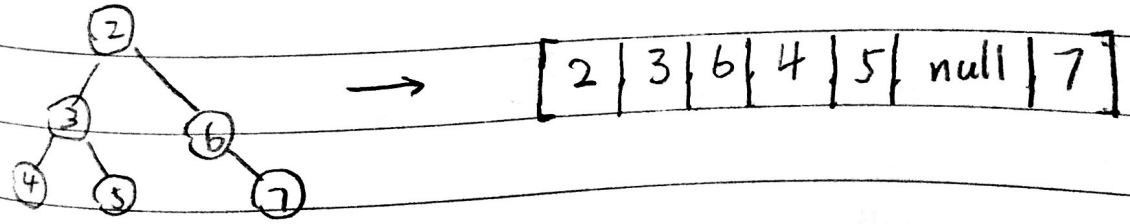
2. The number of $\vec{x}$ is the permutation of $[M]^k$ which is $P_k^{|M|} = \frac{M!}{(M-k)!}$. Assume the $\vec{x}$ is generated randomely from $[M]^k$ and the number of slot is M, the probability of collision is $\frac{M}{P_k^{|M|}} = \frac{(M-k)!}{M!}$,

$\frac{1}{M} = \frac{(M-1)!}{M!}$

Therefore, $\frac{M}{P_k^M} = \frac{(M-k)!}{M} <= \frac{1}{M}$ when $k >= 1$

so H_r is universal.

3. My idea is to firstly find the spanning tree T of the graph G. Secondly, encode the spanning tree T into an array. For example:



Lastly, use the existing methods to hash the array.

4. for each node, run r(4) once (say the result is b)
   this determines how many edges that start from the current node to the other nodes.
   Then randomly select b nodes from other nodes, connect the randomly selected b nodes with the current node

5. When adding a item to the set, pass it to the internal hash function to generate a hash value, the values will be used as a index for the bit array
   The bit array has size of m, which can store m ty of places. Initally the bit array are all zero, when a place added to the set, set bit array[hash(place)] to 1
   For example:
   Initally; bit array is [0, 0, 0, 0, ..., 0]
   Adding place "Apple bee's"
   h("Apple bee's") = 1
   update the bit array to [0, 1, 0, 0, ..., 0]
   when query the set "Is there a restraunt?"
   pass the the query to hash function, h("restraunt") = 1
   check the bit_array[1] .if it is 1 then there is a resta

nearby. Lastly, reset the bit array to 0 for every 30 miles traveled.

6. My idea is to calculate the hash value for each 10 bit array based on base value,

if the right most bit is 1

base value += 1

if the right most bit +1 is 1

base value += 2

⋮

if the left most bit is 1

base value += 10

For example

$[0]^{10}$ = base value = 0

[1, 0, 0, 0, 0, 0, 0, 0, 0, 0] = 0+10 = 10

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1] = 55

[1, 1, 1, 1, 1, 1, 1, 1, 1, 0] = 54

[1, 1, 1, 1, 1, 1, 1, 1, 0, 1] = 53

So in this way, if two bit arrays have small hamming distance, so does their hash values, however, the hash value may not be unique. To have a one to one hash, add the decimal reprentation to the previous has

value.

For example

$[1]^{10} = \underbrace{55}_{\text{hash}} + \underbrace{1024}_{\text{decimal}} = \underbrace{1079}_{\text{final hash}}$

when compare the similarity of the two final hash values, substract the decimal representation.