Problem Set 3.

3-1. Hash practice

(a) $A = [47, 61, 36, 52, 56, 33, 92]$

$h(k) = (10k + 4) \bmod 7$

linked list

$10k + 4$

a b

$A' = [474, 614, 364, 524, 564, 334, 924]$

$A'' = [67, 87, 52, 44, 80, 37, 132]$

$\quad\;\; 5, \;\; 5, \;\; 0, \;\; 6, \;\; 4, \;\; 5, \;\; 0$

~~hash table:~~

67	87	52	44	80	37	132
5	5	0	0	4	5	0

OR

0	1	2	3	4	5	6
~~67~~ ~~52~~ 36				~~80~~ 56	~~67~~	~~144~~ 52
~~132~~ 92					~~87~~	
					~~37~~	

5: 47, 61, 33

store meter number

Memo No.
Date / /

(b) $h(k) = ((10k+4) \bmod c) \bmod 7$, $c>0$

find no collisions occur when inserting the keys from $\mathcal{X}$

$\Rightarrow$ smalles value of c is 13

check:

from code

Problem 3-b

$2n \to n$ rooms 0 to $n-1$,

2n student Id $< u$, $u >> 2n$

hashing Ids to room

$\mathcal{H}$

- k_1, k_2 to one room guarantee
- prove that is ~~no~~ no possible. compute the highest probability they could possibly to be roommates.

(a) $\mathcal{H} = \{h_{ab}(k) = (ak+b) \bmod n \mid a, b \in [0, \cdots n-1]$ and $a \neq 0\}$

it means k_1, k_2 is a pair collision $\Rightarrow$ ~~$k_1 = (ak_2+b) \bmod n$~~

can guarantee k_1 and k_2 to bee roommate

$(ak_1+b)\%n = (ak_2+b)\%n$ $\quad a(k_1-k_2) = q_1 n - q_2 n$

$(ak_1+b) - q_1 n = (ak_2+b) - q_2 n$ $\quad k_1 - k_2 = (\frac{q_1-q_2}{a}) \cdot n$, $k_1 \equiv k_2 \bmod n$

Mo Tu We Th Fr Sa Su

Memo No.

Date / /

$ak_1 + b \equiv ak_2 + b \pmod n,$

$\Rightarrow k_1 \equiv k_2 \pmod n$

(b) $H = \{ h_a(k) = (\lfloor \frac{kn}{u} \rfloor + a) \bmod n \mid a \in \{0, \dots, u-1\} \}$,

set $h_a(k_1) = h_a(k_2)$

$(\lfloor \frac{k_1 n}{u} \rfloor + a) \equiv (\lfloor \frac{k_2 n}{u} \rfloor + a) \bmod n$

$\lfloor \frac{k_1 n}{u} \rfloor \equiv \lfloor \frac{k_2 n}{u} \rfloor \bmod n$

$u >> 2n$

so $\lfloor \frac{k_1 n}{u} \rfloor$ and $\lfloor \frac{k_2 n}{u} \rfloor$ always equal

so k_1 and k_2 are always be roommate is k_1 is approaching to k_2

(c) $H = \{ h_{ab}(k) = ((ak + b) \bmod p) \bmod n$

$p > u$

$h_{ab}(k_1) \equiv h_{ab}(k_2) \bmod n$

$\Rightarrow ((ak + b) \bmod p) \bmod n \equiv ((ak + b) \bmod p) \bmod n$

$\Rightarrow \frac{1}{m}, \Rightarrow \frac{1}{n}$ ~~possi~~ probability

Memo No.

Date / /

P3-3. sort n slice

(a) core identifier $16\lceil \log_4(\sqrt{n})\rceil$

$16\lceil \log_4(\sqrt{n})\rceil$ ASCII characters

$16\lceil \log_4(\sqrt{n})\rceil \times 8$ bit 1 ASCII ~~是~~ 8 bit

$= O(\log n)$ bits

$\log_4\sqrt{n} = \cancel{\frac{1}{2}\log}$ $64\lceil \log_4 n\rceil$

$$2^{16\lceil \log_4(\sqrt{n})\rceil \times 8}$$

$\log_4\sqrt{n} = \frac{1}{2}\log_4 n$ $\quad \log_4 n = \frac{\log_2 n}{\log_2 4} = \frac{1}{2}\log_2 n$

$= \frac{1}{4}\log_2 n$

$\Rightarrow O(\log n)$ bit $\quad w \geq \log n$

max integer $= 2^{16\lceil \log_4(\sqrt{n})\rceil \times 8}$

$= 2^{\lceil \log_2 n\rceil \cdot 64} = n^{64} \cdot \cancel{2^{64}}$

So use radix Sort, runtime $\Theta(n + \log_n n^{64})$

$= \Theta(n)$

(b) 800000 years old sort by age

$[8 \cdot 10^5] < 2^{32}$ use radix Sort $= \Theta(n + \log_n 8 \cdot 10^5)$

$= \Theta(n + 5\log_n 80)$

$= \Theta(n)$

Memo No.

Date / /

(c) $\frac{m}{n^3}$ between 0 and 4

times n^3 $= m$ $m \in [0, 4n^3]$

so use radix sort $u = 4n^3$

runtime $= \Theta(n + n\log_n 4n^3) = \Theta(n)$ ✓

(d) memory

it means we need to compare the slices, the most efficient way is merge sort, $O(n\log n)$ time

P3-4, Pushing Paper

r papers, n boxes n-foot wall

$B = (b_0, \cdots, b_{n-1})$ $b_i \neq b_j, \forall i \neq j$

(b_i, b_j) $|i-j| < n/10 \Rightarrow b_i + b_j = r$

(a) B and r $O(n)$ - time algorithm

whether B contains a close pair

① store B in hash table

② iterate B with i, and find ~~e j~~ $j = r - i$ in B hash table

$\Rightarrow O(n)$

Memo No.
Date /

(b) suppose $r < n^2$ whether B contains
a close pair that fulfills order r.
① radix sort
② two finger algorithm one in first one in last

P 3-5. Anagram Archaeology
string A is an anagram of B if A is
a permutation of the letters in B;
ASCII a to z.

(a) string A, integer k. $d(|A|)$ times
$|B| = (k)$ B in A in $O(k)$ time
~~means~~ it

idea 1: iterate A, from index $i=0$, store
i to i+k in ~~hash table~~ array ~~key~~ ~~value~~ key, take len(A) time ✓
the hash table value store every letter count,
take k^2 time ✓
~~example es~~
$= O(A + k^2) = O(A)$

$O(k)$
~~take~~ input |B|, ~~count~~ every letter times and
iterate with the arry upper ⇒ take $O(A^k)$ time ✗

Mo | Tu | We | Th | Fr | Sa | Su

Memo No.

Date / /

$S_i \in S$

(b) $\underline{T}$ n length-k string $\underline{S = (S_0, \dots, S_{n-1})}$

$0 < k < |T|$ $O(|T| + nk)$

$\Rightarrow$ use (a)