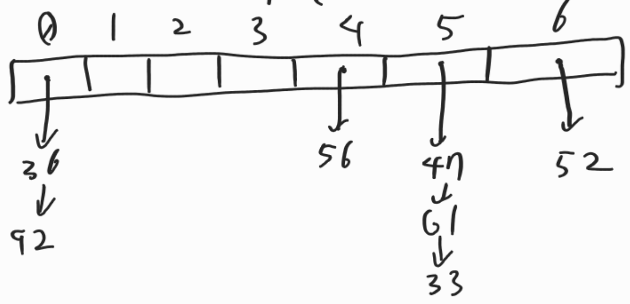
PS 3

3-1.

(a) $h(47)=5$ $h(61)=5$ $h(36)=0$ $h(52)=6$

$h(56)=4$ $h(33)=5$ $h(92)=0$.



(b) $10k+4 = cq_1 + t$ where $0 \le t = 7q_2 + \{0, \dots 6\} < c$

$t(k) = 10k+4 \Rightarrow 474, 614, 364, 524, 564, 334, 924$

$$7n + r \quad \begin{pmatrix} 474 \\ 614 \\ 364 \\ 524 \\ 564 \\ 334 \\ 924 \end{pmatrix} \quad \begin{pmatrix} 7n_1 + 0 \\ 7n_2 + 1 \\ 7n_3 + 2 \\ 7n_4 + 3 \\ 7n_5 + 4 \\ 7n_6 + 5 \\ 7n_7 + 6 \end{pmatrix}$$

We need a bijection.

① For every $k \in A$, $t(k)$ mod c is all different.

②, For every $k \in A$, $h(k)$ is all different.

$c = 13$

3-2

(a) If Rong and Tiri choose numbers multiple of n, they will be roommates.

Let $k_1 = m_1 n$, $k_2 = m_2 n$. Then

$h_{ab}(k_1) = (am_1 n + b) \bmod n = b$

$h_{ab}(k_2) = (am_2 n + b) \bmod n = b$

So $h_{ab}(k_1) = h_{ab}(k_2)$ for any a, b.

(b)

Let $\lfloor \frac{kn}{u} \rfloor = m$ for some $m \in \mathbb{N}^+$. Then

$m \leq \frac{kn}{u} < m+1 \iff \frac{mu}{n} \leq k < \frac{(m+1)u}{n} = \frac{mu}{n} + \frac{u}{n}$. So

$k = \frac{mu}{n} + t$ where $t \in \{i \mid i \in \mathbb{Z},\ 0 \leq i < \frac{u}{n}\} = T$

by assumption, $u > 2n$, so $|T| \geq 2$. We can pick two elements t_i, t_j from T such that

$k_1 = \frac{mu}{n} + t_i \quad k_2 = \frac{mu}{n} + t_j$ where $\lfloor \frac{k_1 n}{u} \rfloor = \lfloor \frac{k_2 n}{u} \rfloor$.

Thus, $h_a(k_1) = (\lfloor \frac{k_1 n}{u} \rfloor + a) \bmod n = (\lfloor \frac{k_2 n}{u} \rfloor + a) \bmod n = h_a(k_2)$.

(c) $k < u < p$, $a, b \in \{0, \ldots, p-1\}$ and $a \neq 0$.

$ak + b < p^2 + p$

3-3.

(a) Radix sort.

First, initialize an static array of size n. And do counting sort by the first character from the rightmost of each slices of core identifier. And repeat counting sort by the second character, and so on, until it reaches the leftmost of the last character. This procedure takes $O(n \log_{(16)}(n))$ time.

(b)

(c)

m = O(n^3).

This implies $\lfloor \log_{(n)}(m) \rfloor \leq 4$. We can use radix sort considering that integer m is key for each of the slices. It takes O(4n) = O(n) time.

(d) The statement implies sorting on a comparison model. Each of the memory cannot be quantified so that we can sort without comparison. The best algorithm for comparison model takes time $\Omega(n\log(n))$, by merge sort, etc.

3-4.

(a)

First, we must sort the boxes by the number of reams each box contains. Any pair of the boxes has equivalent number of reams, and this implies that every bax has its unique key. So we can use DAA sort. This takes O(n) time.

After sorting, follow the algorithm below:

```
Alg find_pair(B', i, j)
input: B' = sort(B). i <= j.
if i > j
   return No such pair
if B'[i] + B'[j] == r
   if |i - j| < n / 10
      return (b_i, b_j) as a connect answer
   else find_pair(B', i + 1, j) or find_pair(B', i, j - 1)
else if B'[i] + B'[j] > r
   find_pair(B', i, j - 1)
else
   find_pair(B', i + 1, j)
```

This algorithm takes also O(n)-time. So entire procedure takes O(n)-time.

3-5.

(a)

Build an array of (|A| - k + 1) length which each slot contains a k-length string. Building takes O(|A|) time. And while traversing the string from the end of the array, record data representing: for some e $\in$ the array, { O_1, ..., O_k } where O_i is the number of elements that

1. back of e
2. contains the same first character of the string $\wedge$ ... second ... $\wedge$ ... $\wedge$ ... i-th ...

(b) |S| = k

If we use the algorithm suggested in (a), we can find A in O(|T| + nk)-time.