

# sparse tables

2019年度前期 高度データ構造

# ハッシュ

---

- ▶  $U = \{1, 2, 3, \dots, u\}$  とする.
- ▶  $S \subseteq U$  とし,  $|S| = n$  とする.
- ▶ 任意の  $x \in U$  について,  
member( $x, S$ ), insert( $x, S$ ), delete( $x, S$ )を  
効率よく処理するためのデータ構造をハッシュという.
- ▶ 他の操作 (min, max, predecessor, successor) は  
必ずしもサポートしなくてよい.



# ハッシュの性能

---

- ▶ ハッシュの領域計算量は  $O(n)$  が望ましい。  
→ member, insert, delete のみサポートすれば良いから。
- ▶ 以降,  $O(n)$  領域のデータ構造のみを考え, ハッシュの性能を  
<memberの最悪時間計算量, insert/deleteの最悪時間計算量>  
で評価する。
- ▶ 連結リスト:  $\langle O(n), O(n) \rangle$
- ▶ チェイン法:  $\langle O(n), O(1) \rangle$
- ▶ オープンアドレス法:  $\langle O(n), O(n) \rangle$
- ▶ AVL木:  $\langle O(\log n), O(\log n) \rangle$
- ▶ q-fast trie:  $\langle O(\sqrt{\log u}), O(\sqrt{\log u}) \rangle$



# 今週のデータ構造

---

▶ **sparse table [Fredman et al., 1984]:  $\langle O(1), O(nu) \rangle$**

▶ **【利点】** member クエリの最悪時間計算量が  $O(1)$ ,  
つまり配列と同程度に速い！

※ 配列は  $O(u)$  領域必要だが, sparse table は  $O(n)$  領域で済む.

**Magic!!**

▶ **【欠点】** insert/delete に最悪で  $O(nu)$  時間かかってしまう.  
したがって, 動的なデータに対してはかなり非効率的.

▶ しかし, 静的なデータに対してはさほど問題にならない.

▶ また, 平均的には  $O(n)$  時間で insert を行うことができる.

---



# 関数 $B_t(w, S, k)$

---

## 定義 1

全体集合を  $U = \{1, 2, \dots, u\}$  とし,  $p = u+1$  を素数とする.  
任意の整数  $k \in U, w \geq n = |S|$  について

$$B_t(w, S, k) = |\{x \in S : (kx \bmod p) \bmod w = t\}|$$

と定義する.

- ▶  $B_t(w, S, k)$  は  $(kx \bmod p) \bmod w = t$  を満たす  $x \in S$  の個数を表す.
- 
- ▶

## $B_t(w, S, k)$ の具体例

---

- ▶  $U = \{1, 2, \dots, 22\}$ ,  $p = 23$ ,  $S = \{4, 7, 8, 17, 20\}$ ,  $n = 5$ ,  
 $w = 5$ ,  $k = 2$  とする.
- ▶  $B_t(5, S, 2) = |\{x \in S : (2x \bmod 23) \bmod 5 = t\}|$
- ▶  $B_0(5, S, 2) = 0$
- ▶  $B_1(5, S, 2) = |\{8, 17\}| = 2$
- ▶  $B_2(5, S, 2) = |\{20\}| = 1$
- ▶  $B_3(5, S, 2) = |\{4\}| = 1$
- ▶  $B_4(5, S, 2) = |\{7\}| = 1$



# Key Lemma 1

---

## 補題 1

以下を満たす  $k \in U$  が存在する.

$$\sum_{t=0}^{w-1} \binom{B_t(w, S, k)}{2} < \frac{n^2}{w}$$

$$B_t(w, S, k) = |\{x \in S \mid (kx \bmod p) \bmod w = t\}|$$



# Key Lemmas 2, 3

---

## 補題 2

以下を満たす  $k \in U$  が存在する.

$$\sum_{t=0}^{n-1} B_t(n, S, k)^2 < 3n$$

## 補題 3

集合  $S' \subseteq U$  のサイズを  $m$  とする.

ハッシュ関数  $h(x) = (k'x \bmod p) \bmod m^2$  が  $S'$  上の単射となる  $k' \in U$  が存在する.



# sparse table の構築 (1)

- ▶ 各  $1 \leq j \leq n$  について,  $S$  の部分集合  $S_j$  を  $S_j = \{x \in S : (kx \bmod p) \bmod n = j\}$  と定義する.

$U = \{1, \dots, 30\}$ ,  $p = 31$ ,  $S = \{2, 4, 5, 15, 18, 30\}$  のとき  $k = 2$  とすると,

2 について,  $(2 \cdot 2 \bmod 31) \bmod 6 = 4$  より  $2 \in S_4$

4 について,  $(2 \cdot 4 \bmod 31) \bmod 6 = 2$  より  $4 \in S_2$

5 について,  $(2 \cdot 5 \bmod 31) \bmod 6 = 4$  より  $5 \in S_4$

15 について,  $(2 \cdot 15 \bmod 31) \bmod 6 = 0$  より  $15 \in S_0$

18 について,  $(2 \cdot 18 \bmod 31) \bmod 6 = 5$  より  $18 \in S_5$

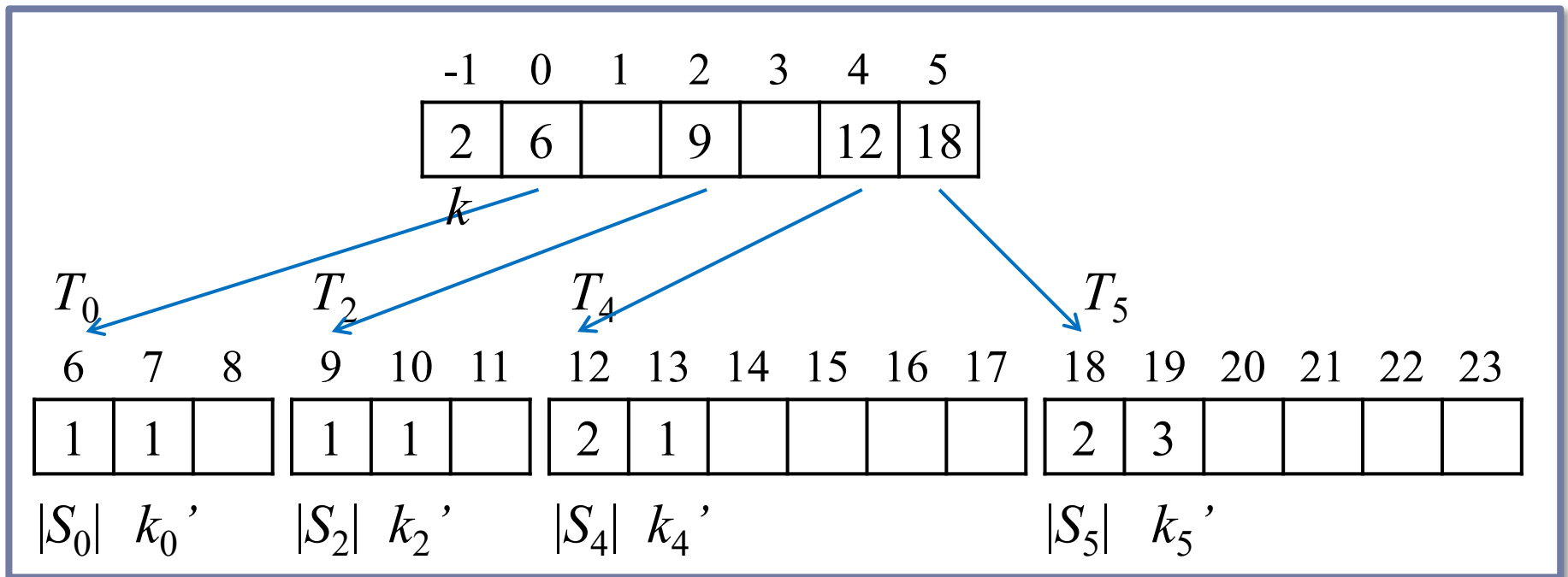
30 について,  $(2 \cdot 30 \bmod 31) \bmod 6 = 5$  より  $30 \in S_5$

よって,  $S_0 = \{15\}$ ,  $S_2 = \{4\}$ ,  $S_4 = \{2, 5\}$ ,  $S_5 = \{18, 30\}$ ,  
 $S_1 = S_3 = \phi$



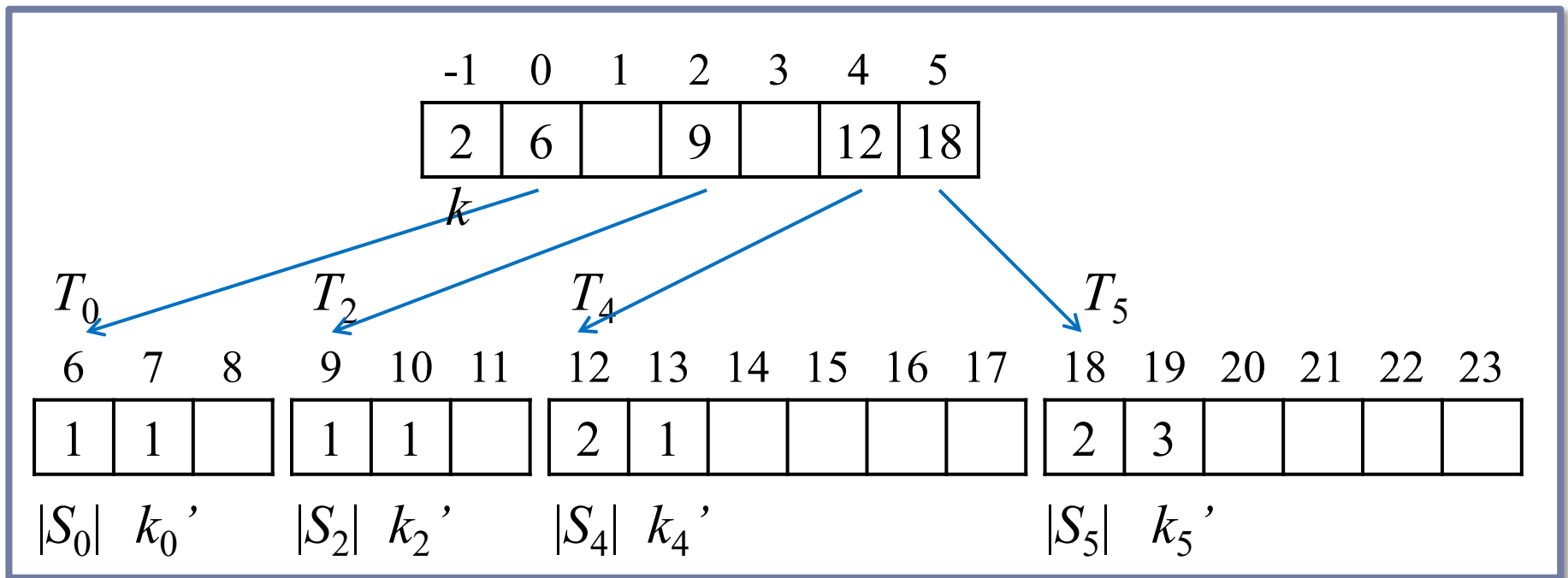
# sparse table の構築 (3)

- ▶  $T_j$  の1番目のセルに  $|S_j|$  を格納する.
- ▶  $T_j$  の2番目のセルに 補題 3 を満たす  $k_j' \in U$  を格納する.



# sparse table の構築 (4)

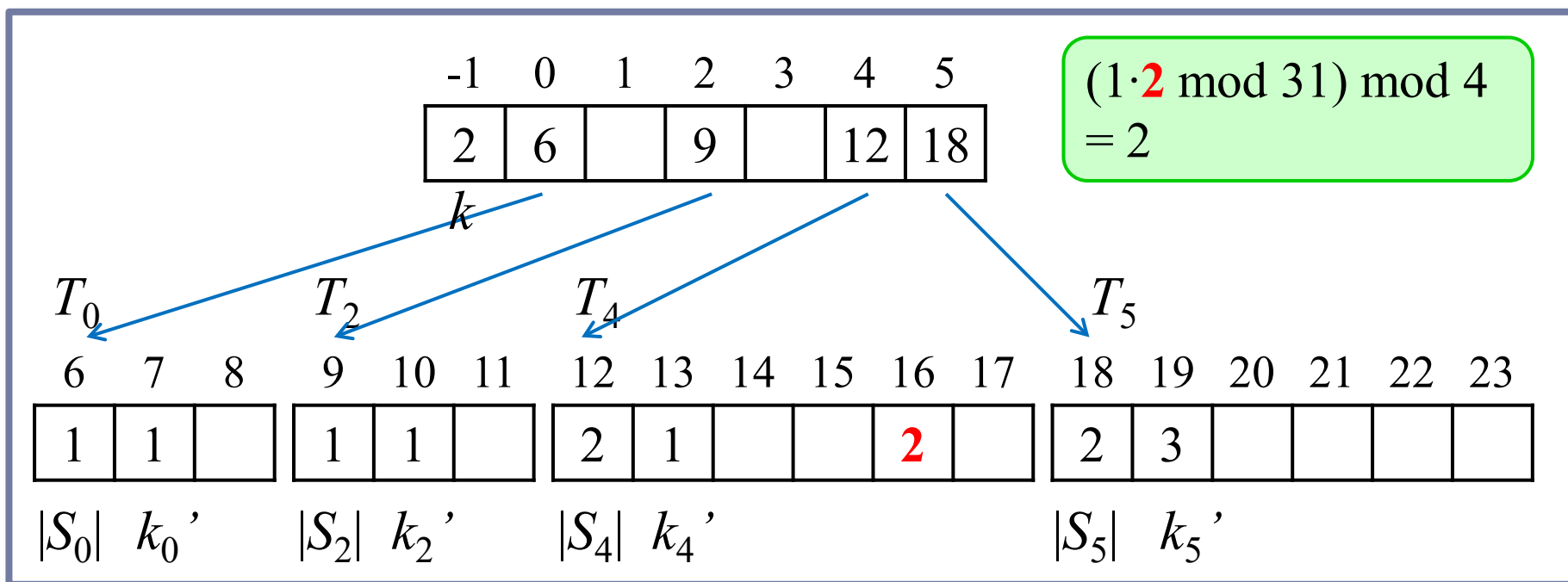
- ▶ 各  $x \in S_j$  を  $T_j$  の  $((k_j' x \bmod p) \bmod |S_j|^2) + 3$  番目のセルに格納する.
- ▶ 補題 3 より, ハッシュ関数  $h(x) = ((k_j' x \bmod p) \bmod |S_j|^2)$  は  $S_j$  上の単射なので, ブロック内で衝突は起きない.



$S = \{2, 4, 5, 15, 18, 30\}$

# sparse table の構築 (4)

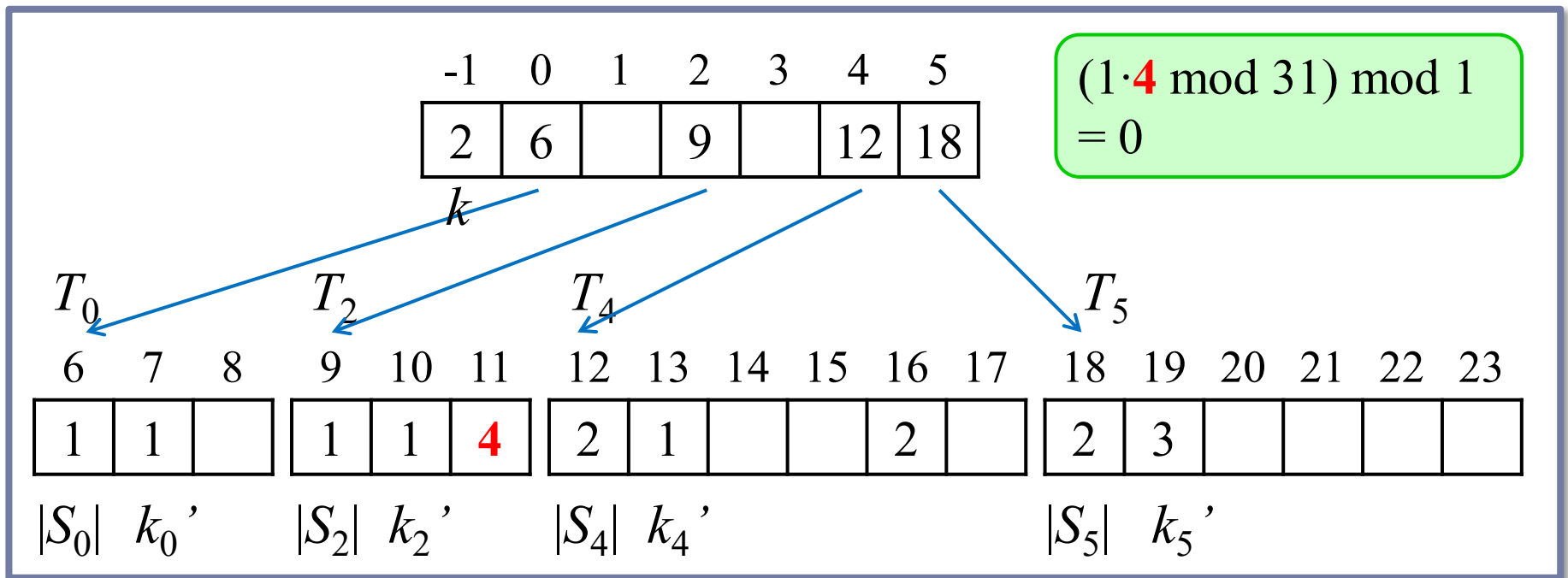
- ▶ 各  $x \in S_j$  を  $T_j$  の  $((k_j' x \bmod p) \bmod |S_j|^2) + 3$  番目のセルに格納する.
- ▶ 補題 3 より, ハッシュ関数  $h(x) = ((k_j' x \bmod p) \bmod |S_j|^2)$  は  $S_j$  上の単射なので, ブロック内で衝突は起きない.



$S = \{2, 4, 5, 15, 18, 30\}$

# sparse table の構築 (4)

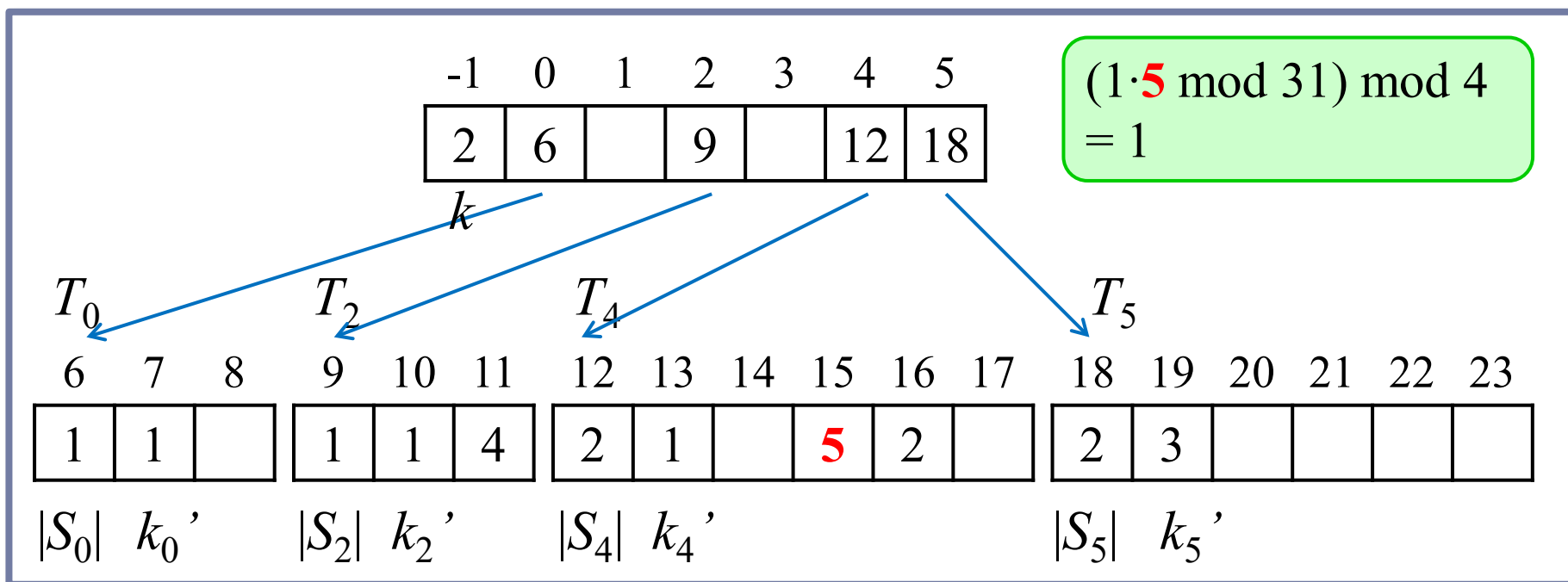
- ▶ 各  $x \in S_j$  を  $T_j$  の  $((k_j' x \bmod p) \bmod |S_j|^2) + 3$  番目のセルに格納する.
- ▶ 補題 3 より, ハッシュ関数  $h(x) = ((k_j' x \bmod p) \bmod |S_j|^2)$  は  $S_j$  上の単射なので, ブロック内で衝突は起きない.



$S = \{2, 4, 5, 15, 18, 30\}$

# sparse table の構築 (4)

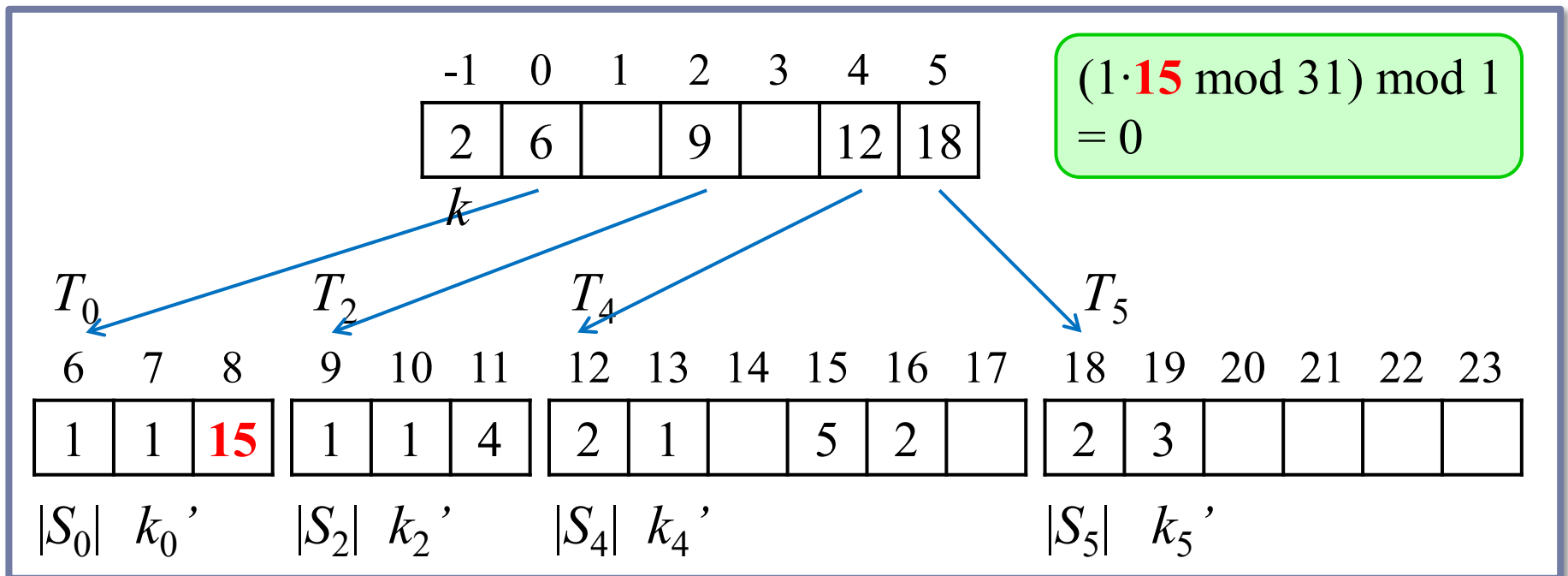
- ▶ 各  $x \in S_j$  を  $T_j$  の  $((k_j' x \bmod p) \bmod |S_j|^2) + 3$  番目のセルに格納する.
- ▶ 補題 3 より, ハッシュ関数  $h(x) = ((k_j' x \bmod p) \bmod |S_j|^2)$  は  $S_j$  上の単射なので, ブロック内で衝突は起きない.



$S = \{2, 4, 5, 15, 18, 30\}$

# sparse table の構築 (4)

- ▶ 各  $x \in S_j$  を  $T_j$  の  $((k_j' x \bmod p) \bmod |S_j|^2) + 3$  番目のセルに格納する.
- ▶ 補題 3 より, ハッシュ関数  $h(x) = ((k_j' x \bmod p) \bmod |S_j|^2)$  は  $S_j$  上の単射なので, ブロック内で衝突は起きない.

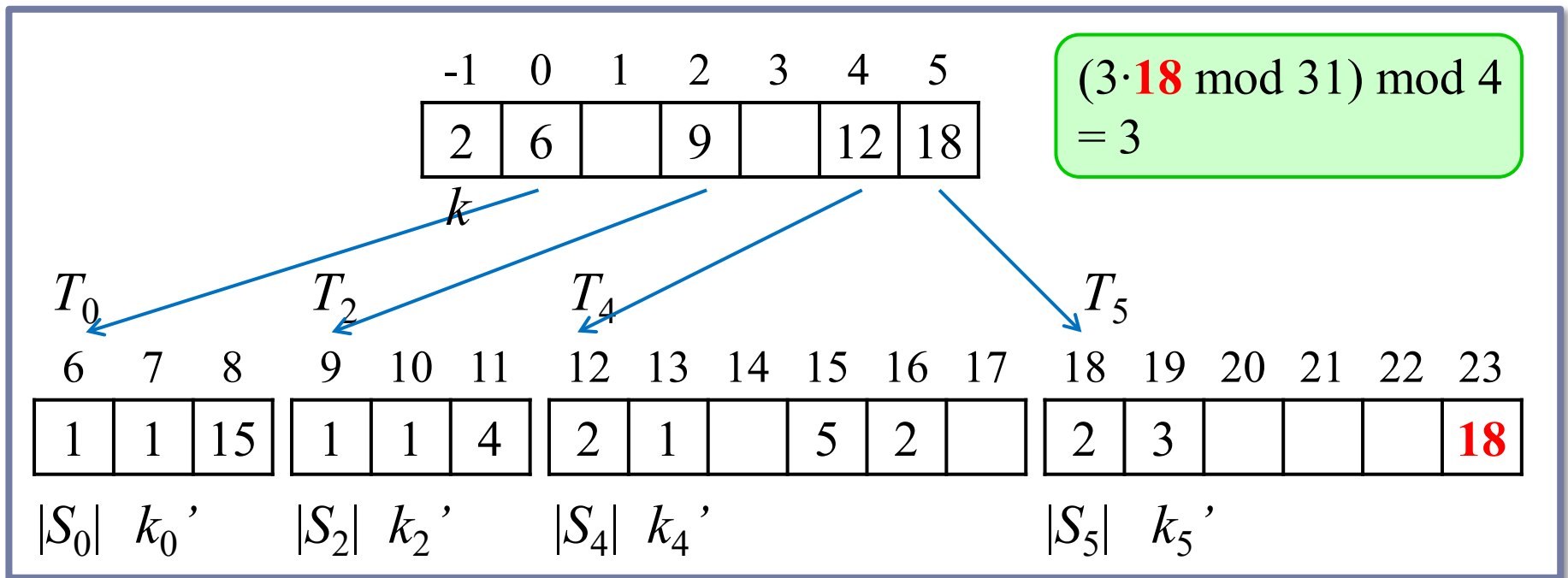


$S = \{2, 4, 5, \mathbf{15}, 18, 30\}$



# sparse table の構築 (4)

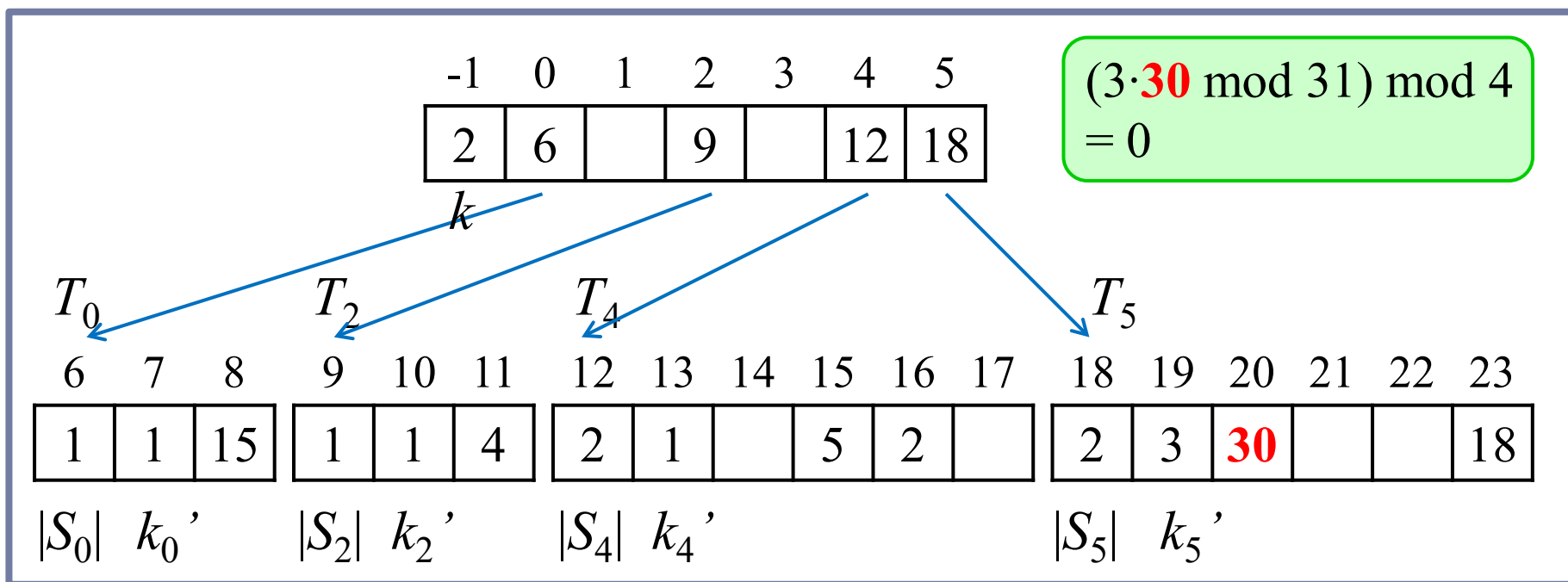
- ▶ 各  $x \in S_j$  を  $T_j$  の  $((k_j' x \bmod p) \bmod |S_j|^2) + 3$  番目のセルに格納する.
- ▶ 補題 3 より, ハッシュ関数  $h(x) = ((k_j' x \bmod p) \bmod |S_j|^2)$  は  $S_j$  上の単射なので, ブロック内で衝突は起きない.



$S = \{2, 4, 5, 15, 18, 30\}$

# sparse table の構築 (4)

- ▶ 各  $x \in S_j$  を  $T_j$  の  $((k_j' x \bmod p) \bmod |S_j|^2) + 3$  番目のセルに格納する.
- ▶ 補題 3 より, ハッシュ関数  $h(x) = ((k_j' x \bmod p) \bmod |S_j|^2)$  は  $S_j$  上の単射なので, ブロック内で衝突は起きない.



$S = \{2, 4, 5, 15, 18, 30\}$

# sparse table の領域計算量

---

## 定理 1

サイズ  $n$  の集合  $S$  に対する sparse table の領域計算量は  $O(n)$  である。

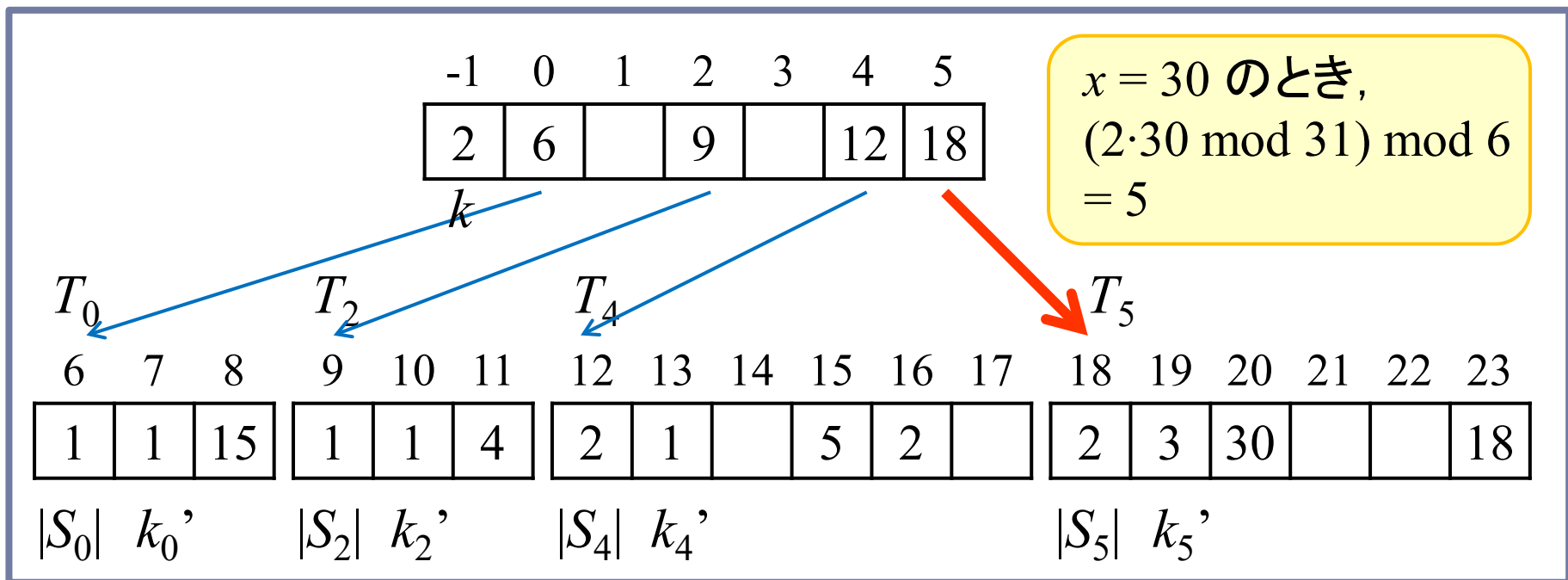
- ▶ 各ブロックへのポインタが  $n$  個（空ブロック分を含む）。
- ▶ 補題 2 より, 各ブロックの合計サイズは

$$\sum_{j=0}^{n-1} (|S_j|^2 + 2) = 2n + \sum_{j=0}^{n-1} |S_j|^2 < 5n$$

- ▶ よって, sparse table のサイズは  $O(n)$  である。
-

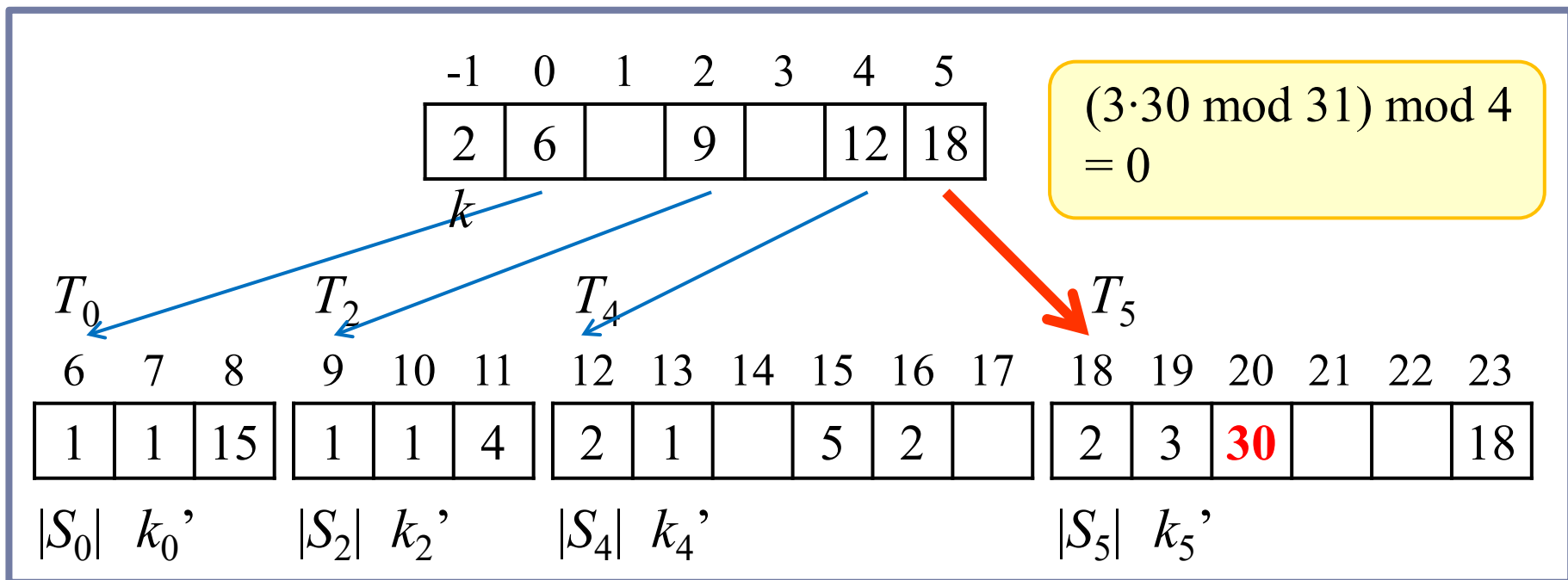
# sparse table を用いた memberクエリ

- ▶  $\text{member}(x, S)$  を計算したい.
- ▶  $k = T[-1]$  とし,  $j = (kx \bmod p) \bmod n$  とする.
- ▶  $T[j]$  の値を利用して, ブロック  $T_j$  にアクセスする.



# sparse table を用いた memberクエリ

- ▶  $|S_j|$  と  $k_j'$  の値を読み込む.
- ▶ ハッシュ値  $h(x) = (k_j' x \bmod p) \bmod |S_j|^2$  を計算し,  $T_j$  の  $h(x) + 3$  番目のセルの値が  $x$  であるか調べる.



# memberクエリの正当性と計算時間

---

## 定理 2

sparse table を用いることで、  
 $\text{member}(x, S)$  を  $O(1)$  時間で計算できる。

- ▶ 【時間計算量の証明】
  - ▶ 配列  $T$  の要素の参照回数は  $O(1)$ .
  - ▶ 剰余 (mod) の計算回数も  $O(1)$ .
  - ▶ よって、最悪時でも  $O(1)$  時間で計算できる。

# memberクエリの正当性と計算時間

---

## 定理 2

sparse table を用いることで、  
 $\text{member}(x, S)$  を  $O(1)$  時間で計算できる。

### ▶ 【正当性の証明】

1.  $x \in S$  のとき, sparse table の構築方法より自明.
2.  $x \notin S$  のとき, 以下の2つの場合が存在する:
  - A)  $S_j = \phi$  のとき
  - B)  $S_j \neq \phi$  のとき



# memberクエリの正当性と計算時間

---

## 定理 2

sparse table を用いることで,  
 $\text{member}(x, S)$  を  $O(1)$  時間で計算できる.

### ▶ 【正当性の証明】

A)  $S_j = \phi$  のとき

この場合, 対応するブロック  $T_j$  が存在しない.

命題「 $x \in S$  ならば  $T_j$  が存在する」が真なので,  
その対偶「 $T_j$  が存在しないならば  $x \notin S$ 」も真.





# memberクエリの正当性と計算時間

---

## 定理 2

sparse table を用いることで、  
 $\text{member}(x, S)$  を  $O(1)$  時間で計算できる。

### ▶ 【正当性の証明】

B)  $S_j \neq \phi$  のとき

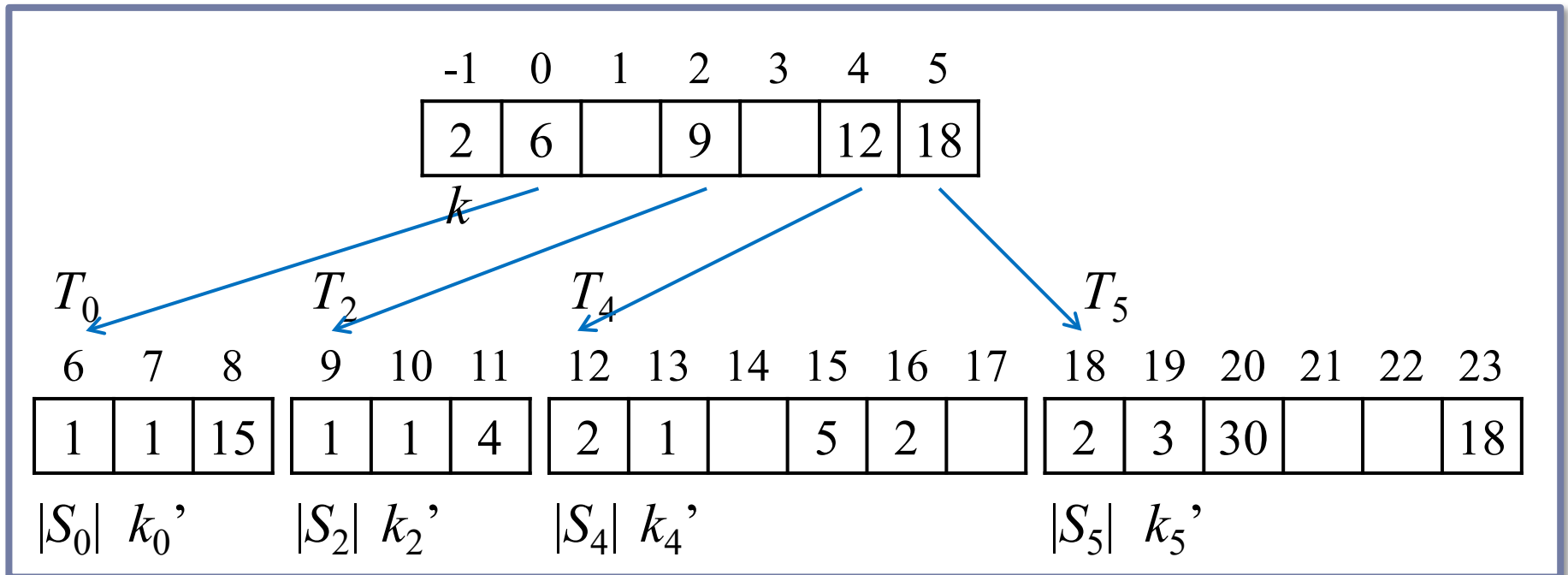
- i. ブロック  $T_j$  の対応するセルが nil のとき  
命題「 $x \in S$  ならば  $T_j$  の対応するセルに  $x$  がある」が真なので、その対偶も真。
- ii. ブロック  $T_j$  の対応するセルに  $y (\neq x)$  があるとき  
上の場合と同様。



# 練習問題

▶ 以下の sparse table ( $p = 31$ ) を用いて:

1.  $\text{member}(5, S)$  を計算せよ.
2.  $\text{member}(24, S)$  を計算せよ.
3.  $\text{member}(23, S)$  を計算せよ.



# sparse table の構築時間

---

## 定理 3

サイズ  $n$  の集合  $S$  に対する  
sparse table を  $O(nu)$  時間で構築可能.

1. まず,  $k = 1, 2, \dots, u$  と順に試していき,  
補題 2 を満たす  $k$  を見つける.
2. ステップ 1 で見つかった  $k$  を用いて,  $S_j$  を計算する.  
各  $S_j$  について, 補題 3 を満たす  $k_j'$  を見つける.

ステップ 1, 2 共に  $O(nu)$  時間でできる.

---

# sparse table に対する insert/delete 操作

---

## 定理 3

サイズ  $n$  の集合  $S$  に対する  
sparse table を  $O(nu)$  時間で構築可能.

- ▶ insert を行う場合, 要素数が  $n+1$  になるので, sparse table を一から構築する必要がある.
- ▶ したがって, insert/delete にも  $O(nu)$  時間かかってしまう.



# Key Lemma 1

---

## 補題 1

以下を満たす  $k \in U$  が存在する.

$$\sum_{t=0}^{w-1} \binom{B_t(w, S, k)}{2} < \frac{n^2}{w}$$

$$B_t(w, S, k) = |\{x \in S \mid (kx \bmod p) \bmod w = t\}|$$



## 補題1の証明 [1/5]

---

- ▶ 次式を満たす  $k$  と  $x, y \in S$  (ただし  $x \neq y$ ) を考える.

$$(kx \bmod p) \bmod w = (ky \bmod p) \bmod w$$



## 補題1の証明 [1/5]

---

- ▶ 次式を満たす  $k$  と  $x, y \in S$  (ただし  $x \neq y$ ) を考える.

$$\frac{(kx \bmod p)}{X} \bmod w = \frac{(ky \bmod p)}{Y} \bmod w$$

- ▶  $X = kx \bmod p$ ,  $Y = ky \bmod p$  とする.  
 $p (= u+1)$  が素数で, かつ  $k, x, y < p$  より,  $X \neq Y$  である.

- ▶ よって, ある  $0 \leq d < w$  と  $i \neq j$  について,

$$X = iw + d \text{ かつ } Y = jw + d \quad \cdots (1)$$

- ▶ 一般性を失うことなく,  $i > j$  と仮定する.
- 



## 補題1の証明 [2/5]

---

- ▶ 式(1) より,  $X = Y + (i - j)w$  なので,

$$kx \bmod p = ky \bmod p + (i - j)w \quad \cdots (2)$$

- ▶ ここで,

$$0 \leq kx \bmod p = ky \bmod p + (i - j)w < p \quad \cdots (3)$$

- ▶ 一方で,

$$0 \leq ky \bmod p < p \quad \cdots (4)$$

- ▶ 式(3), (4) より,

$$kx \bmod p = (ky + (i - j)w) \bmod p \quad \cdots (5)$$

---





## 補題1の証明 [3/5]

---

▶ いま式(5)  $kx \bmod p = (ky + (i-j)w) \bmod p$  が成立.

▶ ある  $0 \leq c < p$  と  $l, r \geq 0$  について,

$$kx = lp + c \text{ かつ } ky + (i-j)w = rp + c \quad \cdots (6)$$

▶ 式(6) より

$$k(x-y) = (l-r)p + (i-j)w \quad \cdots (7)$$

▶ 式(3) より  $(i-j)w < p$  なので,

式(7) より  $k(x-y) \equiv (i-j)w \pmod{p}$ , すなわち

$$k \equiv (i-j)w(x-y)^{-1} \pmod{p} \quad \cdots (8)$$

▶ 整数  $(x-y)$  の  $p$  を法とする multiplicative inverse

# modular multiplicative inverse

---

## 定義 2

整数  $m$  を法とする 整数  $a$  の multiplicative inverse  $a^{-1}$  とは  $az \bmod m = 1$  を満たす最小の正整数  $z$  のことをいう.

- ▶ 例) 11 を法とする 3 の multiplicative inverse は,  $3z \bmod 11 = 1$  を満たす最小の正整数  $z$  である.  
→  $z = 4$
- 
- ▶

## 補題1の証明 [4/5]

---

- ▶ いま式(8)  $k \equiv (i-j)w(x-y)^{-1} \pmod{p}$  が成立.
- ▶  $x, y \in S$  がとり得る値の組み合わせは  $n^2$  未満 ( $n = |S|$ ).
- ▶  $0 < (i-j)w < p$  より,  $i-j$  のとり得る値は高々  $(p-1)/w$  種類.
- ▶ よって, 式(8)を満たす  $(k, x, y)$  の種類は  $(p-1)n^2/w$  未満. つまり, 次式が成り立つ.

$$\sum_{k=1}^{p-1} \sum_{t=0}^{w-1} \binom{B_t(w, S, k)}{2} < \frac{(p-1)n^2}{w} \quad \dots (9)$$

---



## 補題1の証明 [5/5]

---

▶ いま式(9)  $\sum_{k=1}^{p-1} \sum_{t=0}^{w-1} \binom{B_t(w, S, k)}{2} < \frac{(p-1)n^2}{w}$  が成立.

▶ よって, ある  $k \in \{1, 2, \dots, p-1 = u\} = U$  について

$$\sum_{t=0}^{w-1} \binom{B_t(w, S, k)}{2} < \frac{n^2}{w}$$

が成立する.

【証明終】



# Key Lemma 2

---

## 補題 2

以下を満たす  $k \in U$  が存在する.

$$\sum_{t=0}^{n-1} B_t(n, S, k)^2 < 3n$$



## 補題 2 の証明

---

$$\begin{aligned} \sum_{t=0}^{n-1} \binom{B_t(n, S, k)}{2} &= \sum_{t=0}^{n-1} \frac{B_t(n, S, k)(B_t(n, S, k) - 1)}{2} \\ &= \frac{1}{2} \left( \sum_{t=0}^{n-1} B_t(n, S, k)^2 - \sum_{t=0}^{n-1} B_t(n, S, k) \right) \\ &= \frac{1}{2} \left( \sum_{t=0}^{n-1} B_t(n, S, k)^2 - n \right) < \frac{n^2}{n} = n \end{aligned} \quad \text{(不等式は補題 1 より)}$$

よって  $\sum_{t=0}^{n-1} B_t(n, S, k)^2 < 3n$

---



# Key Lemma 3

---

## 補題 3

集合  $S' \subseteq U$  のサイズを  $m$  とする.

ハッシュ関数  $h(x) = (k'x \bmod p) \bmod m^2$  が  $S'$  上の単射となる  $k' \in U$  が存在する.



## 補題 3 の証明

---

補題 1 において,  $w = m^2$ ,  $k = k'$  とおくと

$$\sum_{t=0}^{m^2-1} \binom{B_t(m^2, S', k')}{2} < 1$$

すべての  $t = 0, \dots, m^2 - 1$  について,  $B_t(m^2, S', k')$  の要素を 2つ選ぶ組み合わせの合計が 1 未満.

つまり, 任意の  $t$  について,  $B_t(m^2, S', k') \leq 1$

---





# 予告 1

---

## ▶ 次回のデータ構造

### ▶ **Cuckoo hash [Pagh & Rodler, 2004] :**

member クエリを 最悪  $O(1)$  時間で実現しつつ,  
insert 操作を 平均ならし  $O(1)$  時間で行える  
 $O(n)$  領域のハッシュデータ構造.

