מבנה הנתונים 14

משפחה אוניברסלית של פונקציות

נבחר B מספר ראשוני

נפרק את x (מפתח) ל $r+1$ בתים $x=(x_0,x_1,\dots,x_r)$

בדרישה: כל byte x_i קטן מ B

פונקציה: יהי $(a_0,\dots,a_r)$ רצף של מספרים הנבחרים באקראי מתוך $\{0,1,\dots,B-1\}$

ונגדיר $h_a \in H$

$$h_a(x)=\left(\sum_{i=0}^{r} a_i x_i\right) \bmod B$$

$$H=\bigcup_a \{h_a\} \qquad |H| = B^{r+1}$$

<u>טענה:</u> H היא משפחה אוניברסלית של פונקציות Hash

הוכחה: יהיו $x\neq y$ שני מפתחות

נניח בה"כ $x_0 \neq y_0$ (אחרת נחליף באינדקס המתאים במקום x_0)

נקבע את הערכים של $a_1\dots a_r$

<u>טענה:</u> קיים לכל היותר רק אחד של a_0 שעבורו ש- x,y יתנגשו $a_1\dots a_r$.

$$a_0x_0+\sum_{i=1}^{r} a_i x_i=\sum_{i=0}^{r} a_i x_i=\sum_{i=0}^{r} a_i y_i=a_0y_0+\sum_{i=1}^{r} a_i y_i \pmod B$$

$$a_0(x_0-y_0) = -\sum_{i=1}^{r} a_i(x_i-y_i) \pmod B \text{ ולכן}$$

מכיוון ש $x_0 \neq y_0$ ו B ראשוני, x_0-y_0 הפיך מודולו B, לכן

ההופכי מודולו B של x_0-y_0

$$-(x_0-y_0)^{-1}\sum_{i=1}^{r} a_i(x_i-y_i) = a_0 \equiv \cancel{-\frac{\sum_{i=1}^{r} a_i(x_i-y_i)}{(x_0-y_0)} \pmod B}$$

ולכן יש a_0 יחיד

נניח שלכולם פונק' ה ננתח שקולות שווה ב-2 מקומות.
נקבל סכומים (שתקבל שווים שלא שווים שמקבלים β)

ולכן יש שיוויון ש a_0 ל $-\sum_{i=1}^{k}(x_i+y_i)$

מאין שני פתרונות לכן השוויון מתקיים,

לכן לכל ~~...~~ נתונים של $a_1 \dots a_r$

יש פונקציה יחידה שמפרידה את x ו- y.

לכן יש B^r פונקציות שמפרידות את x ו-y

מתוך B^{r+1} פונקציות ולכן

$$P\left(\text{מפרידה בין x ל y}\right) = \frac{B^r}{B^{r+1}} = B$$

זכרון, להשוות שבערך מקודם, אוניברסלי פשוט.

שיטות קידוד

Hash סגור Open ADDRESSING

B סלים

N מיקומים (כרגע בקבלה)

פתרון כאילו: Hash פתוח

הגדרה: יש לנו התנגשות, סכום לשטף להתנגשויות שונות

אם יש אפשרות של y, נלך מספר ...

מתפלג ל- $\frac{y}{B}$ מקומות

כדאי: קידוד ... re-Hsh ...

ההסתברות להתנגשות היא $\frac{N}{B}$

ההסתברות לפחות 2 התנגשויות $\frac{N}{B}\cdot\frac{N-1}{B-1}$

ההסתברות לפחות i התנגשויות: $\frac{N(N-1)\cdots(N-i+1)}{B(B-1)\cdots(B-i+1)}$

עוד

B, N נתונים

לפחות i התנגשויות $\approx (N/B)^i$

תוחלת כמה ניסיונות

$$1 + \sum_{i=1}^{\infty} i \cdot P(\text{בדיוק } i \text{ התנגשויות}) =$$

טריק

$$\sum_{i=1}^{\infty} i\, P(i \text{ התנגשויות}) = \sum_{i=1}^{\infty} P(\text{לפחות } i \text{ התנגשויות})$$

$$1 + \sum_{i=1}^{\infty} P(\text{לפחות } i \text{ התנגשויות}) \le 1 + \sum_{i=1}^{\infty} (N/B)^i = 1 + \frac{\frac{N}{B}}{\frac{N-1}{B}} = \frac{B}{B-N}$$

ניתן עדיין לשפר את ה- $\left(\frac{N}{B}\right)^i$

$$N(N-1)\cdots(N-i+1) / B(B-1)\cdots(B-i+1)$$

נקבל לאחר חישוב מסובך יותר

$$\frac{B+1}{B+1-N}$$

כאשר נוסיף לטבלה את המפתחות $1,2,\dots,M$

Amortized

$$\frac{1}{M} \cdot \sum_{N=0}^{M-1} \frac{B+1}{B+1-N} \approx \frac{1}{M}\int_0^{M-1} \frac{B}{B-x}\,dx = \frac{B}{M}\left(\ln B - \ln(B-M+1)\right) =$$

$$= \frac{B}{M} \ln\left(\frac{B}{B-M+1}\right)$$

אם נמלא את כל הטבלה כלומר B מפתחות

$$\frac{B}{B} \ln\left(\frac{B}{1}\right) = \ln B$$

זה לא Amortized $O(1)$!

למשל אם הגענו ל- 90%. $m=(0.9B)$

נקבל $\frac{9}{10}\ln 10 = O(1)$ Ammortized.

מציאה תמיד:

מציאה שלא קיים כמו הכנסה

מציאה שקיים כמו כשהכנסנו עד כה:

קיים / לא קיים

$$\frac{B+1}{B+1-M} > \frac{B}{M}\ln\frac{B}{B-M\pm 1}$$

אם M כמו α אחוז מ B תוחלת הכנסה היא

$$\frac{1}{\alpha}\ln\frac{1}{1-\alpha}$$

למשל אם B מלא ב 90% מ 1000

אז כמו הכנסה 90% ~ 10^{800}.

שיטות Rehash אחרות:

1) לינארי $h_i(x) = h(x)+i \mod B$ בעייה כי כל מגיע לאזור הקבוצה והקבוצה גדלה,

2) $h_i(x)=h(x)+ci \pmod B$ לא עוזר למנוע יצירה של קבוצות גדולות, מצב שלא נרצה, לכן כדי!

אם $\gcd(B,c)=1$ אז לא נעבור על כל התאים

צריך שלכל המקומות יהיה סיכוי שווה כמו פעם אחת

3) יותר טוב: $h_i(x)=h(x)+d_i \mod B$

כאשר d_i הוא פרמוטציה של $1,2,\dots,B-1$

• אבל כל מי שמגיע עם אותו $h(x)$ יעבור אותו מסלול (בעייה)

כלומר כל מי שמגיע ל $h(x)$ יעבור אותו מסלול בדיוק.

רב - מנגנון

עלות מחזור - קריאה.

עלות כתיבה:

כאשר $N \geq 2B$ כמות לכתיבת את המערך

ולמימוש, ולמיינים את כולם

עלות מיזוג.

אם $N < 0.9B$ ~~מחוק~~ לכתוב את המערך

~~ולמיינים~~ ולמיינים מוכן.

שיעור 9

HASH, טבלת:

המטרה לממש מילון (חיפוש, הוספה, מחיקה)

סיבוכיות: $O(1)$ ממוצע

~~ממוצע~~ $O(\log N)$ W.C.

הגדרה לפונקציה הבינה.

ברוב המקרים אנו משתמשים ב hash לא טוב,

באופן "hash לא טוב".

Perfect hashing

נתונה קבוצה $S \subseteq U$ כאשר $|S|=n$

S קבוצה סטטית מתוך U נתון.

נרצה לבנות טבלת hash בגודל $O(n)$ פעולת חיפוש

שאלה: האם $x \in S$ בזמן קבוע $O(1)$

המטרה לבנות hash כזה:

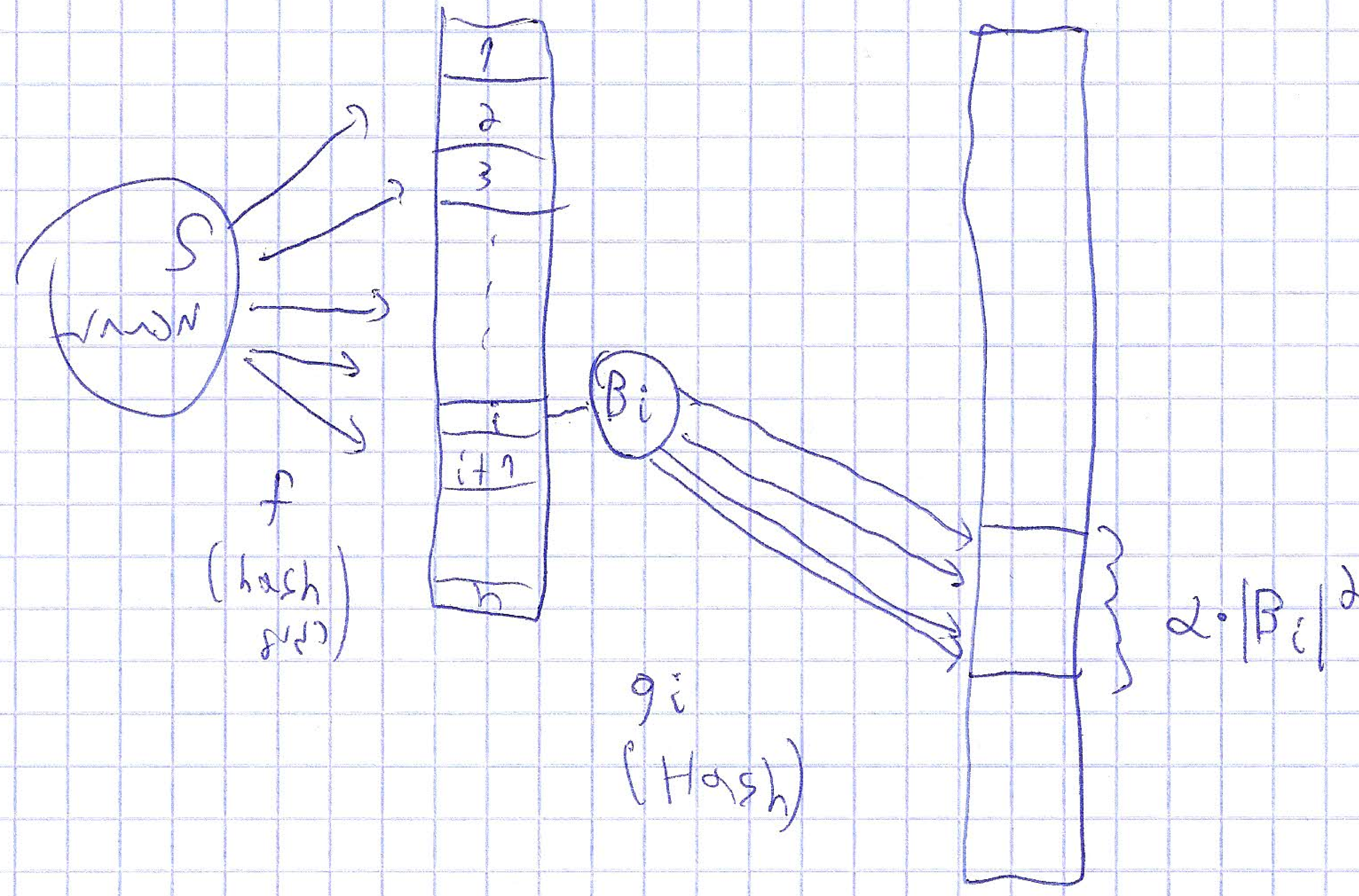
• עלות בנייה

• זמן חיפוש במקרה ה W.C.

• מבנה זמן בנייה

• מקום $O(n)$

טבלה דו שלבית:



עקרונות: 1) f ממפה מפתחות לטבלה קטנה בגודל n

המטרה לבנות קבוצות עם יותר מפתחות אחת

2) מבלה מה של פתרונות g_i שלמה

מוצא ל[illegible] ~~[illegible]~~ בפתרון את הערכים.

ניתוח:

1) זמן בניית T בזמן תוחלתי $O(n)$

צריך למצוא f כך שכל B_i שמכיל

2) מציאת (למצוא) g_i שימוש B_i ל $\alpha|B_i|^2$

כל [illegible]

סה"כ לשמור - [illegible]

[illegible] פתרון ל 1) ו- 2) [illegible].

[illegible] למצוא [illegible]

[illegible] בזמן [illegible]

$b_i = |B_i|$

$$\sum b_i^2 \le n \cdot \beta$$

כיצד ניתן, באופן יעיל, לבחור f אוניברסלית מ משפחה אוניברסלית?

אבחנה:

1) אם נבחר $f: U \to [1 \dots n]$ אוניברסלית אז

מכל זוג אובייקטים תוחלת מספר ההתנגשויות של מפתח בודד (בערך) $\frac{n-1}{\beta}$

לכן בממוצע יש n מפתחות ל $n-1$

$\frac{n-1}{n}$ תוחלת מס' ההתנגשויות של מפתח בודד

לכן תוחלת מספר ההתנגשויות של כל המפתחות

$$n\left(\frac{n-1}{n}\right) = n-1$$

אבל $\#\text{התנגשויות} = \sum \underbrace{b_i}_{\text{מס' ההתנגשויות}} (b_i - 1) = \sum b_i^2 - \sum b_i$

אבל $\sum b_i = n$ כי כל מפתח נכנס באחד מה b_i כלשהו

אז

$$\sum b_i^2 = n + \#\text{collisions}$$

$$E\left[\sum b_i^2\right] = n + E[\#\text{collisions}] = n + n - 1 \le 2n$$

2) לפי מרקוב: אם x מ"מ לא שלילי:

$$P(X \ge t) \le E[X]/t \Rightarrow P(X \ge \tau E[X]) \le \frac{1}{\tau}$$

לכן

$$P\left(\sum b_i^2 \ge 2E\left[\sum b_i^2\right]\right) \le \frac{1}{2}$$

$$P\left(\sum b_i^2 \ge 4n\right) \le \frac{1}{2}$$

נבחר f אוניברסלית, בתוחלת תוך 2 ניסיונות נמצא f טובה.

מסקנה: האלגוריתם של בניית [illegible] f

בוחר p

$$\sum b_i^2 < 4n$$

לאחר שבחרנו פונקציה f

זמן הריצה של חישוב f [illegible] בתוחלת

f [illegible] $O(n)$

[illegible] סכום הריבועים

בחירת g_i

נסמן $b_i = |B_i|$ [illegible]

נבחר [illegible] טווח

$$[1, 2, \ldots, \alpha b_i]$$

נבחר $\alpha = 2$

תוחלת מס' ההתנגשויות של איבר בודד

(*) $$E(C_x) = \frac{b_i - 1}{\alpha b_i^2} < \frac{1}{\alpha b_i^2}$$

תוחלת מס' ההתנגשויות של כל האיברים

$$\frac{b_i(b_i-1)}{\alpha b_i^2} < \frac{1}{\alpha} = \frac{1}{2}$$

נעזר במרקוב ו- *

$$P(C_x \geq 1) \leq E(C_x) \leq \frac{1}{\alpha b_i}$$

סה"כ ההסתברות שיש התנגשות היא

[illegible] Boole [illegible]

$$P(C_{x_1} \cup C_{x_2} \cup \ldots \cup C_{x_{b_i}}) \leq P(C_{x_1}) + \ldots + P(C_{x_{b_i}}) = \frac{b_i}{\alpha b_i} = \frac{1}{\alpha} = \frac{1}{2}$$

סה"כ בהסתברות $\frac{1}{2}$ אין בכלל התנגשויות !!!

[illegible]

לכן [illegible] [illegible] עד ש[illegible] g_i [illegible]

כמה זמן לבנות φ_i :

בדיקת שוויון [illegible] בגודל $2 B_i^2$

יש לפחות - $2 B_i^2$ פעולות

וגם $\sum b_i^2 \le 2ch$

(הגדרנו קבוצה של f שמקיים את 5(ו))

~~כל [illegible] של φ_i לוקח $O(h)$~~

בערך שלב לבנות את כל ה φ_i בעזרת הפונקציות הקודמות $O(h)$

וכיוון [illegible] קבוע [illegible] (2)

שלב $O(h)$ סך הכל.

צריך להוכיח, עבור ה W.C. ב

[illegible]

נניח [illegible] לכל פונקציה חשבה שאנחנו [illegible]

יש h^h פונקציות. [illegible]

נניח ויש [illegible] h^h

[illegible]

ב W.C. [illegible] $\Omega(h^h)$

[illegible]

$$1+2+3+\dots+h^h-1+h^h = \Omega((h^h)^2) = \Omega(h^{2h})$$

~~[illegible] $O(h)$ [illegible]~~

[illegible]

אבל לא נכון:

[illegible]

$\Omega((h)^{\frac{h}{2}})$

[illegible]

$\Omega((h^{\frac{h}{2}})^2)$

[illegible]

$\Omega(h^{\frac{h}{2}+1})$

לא מדויק

[illegible]

$[1,\dots,h]$

$[1,h]$

[illegible]

$\frac{1}{h}$

Union find

מבנה נתונים שאומר לאיזה קבוצה [illegible]

פעולות

make(x) — יצירת קבוצה חדשה עם x

Union(x,y) — איחוד הקבוצות ~~קיימות~~ שמכילות את x ו-y

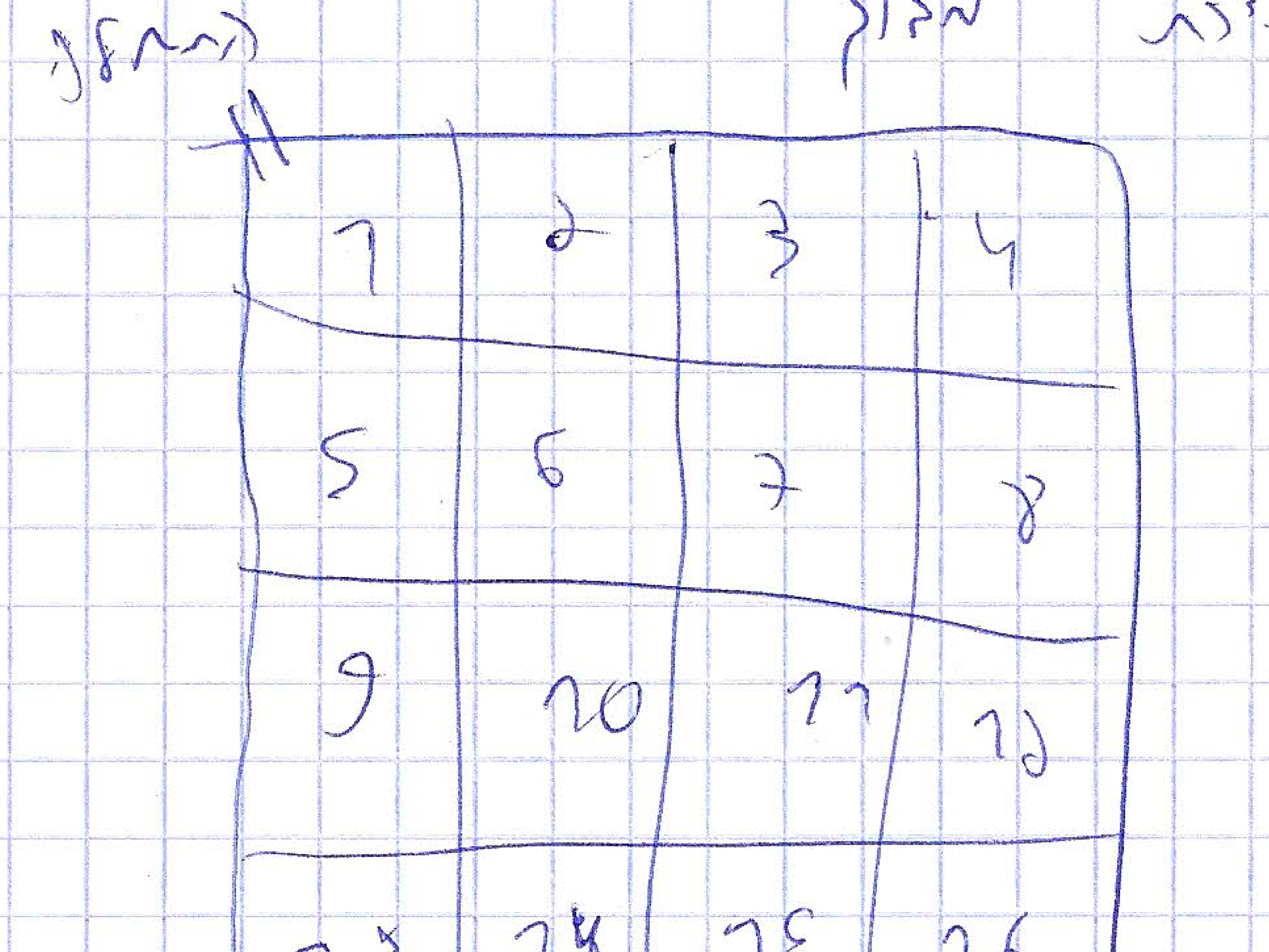
Find(x) — החזרת מזהה שמייצג את הקבוצה שמכילה את x.

make $O(1)$

union $O(\alpha(n))$

find $O(\alpha(n))$

דוגמה לכך ~ יצירת מבוך



make(1), make(2) ... make(16)

find(6) = find(7)

union(6,7)

שונים ~ הורסים קיר

של התאים השונים