

موضوع: ادامه بحث در هم سازی

x value

✓ insert(key, value)  
✓ search(key)  
✓ delete(key)

یادآوری: دیکشنری: (key, value)

نیاز به راه حل سریع داریم.  
پایه سازی ها }  
آرایه مرتب  
نامرتب  
BST متوازن  $\leftarrow \log n$

$A[i]$

$(x, i) \leftarrow$

آدرس دهی مستقیم: مثلاً اگر کلیدها در بازه  $[0 \dots 9999]$  باشد.

زمان:  $O(1)$

\* مشکل: زمانی که بازه کلیدها به نسبت تعداد عناصر خیلی کم باشد  
مانند شماره دانشجویی.

بازه کلید بزرگ باشد.

جدول در هم سازی (Hash Table):

مجموعه کلید:  $U$  اندازه  $U$  بسیار بسیار بزرگ است.  $U$  شامل اعداد

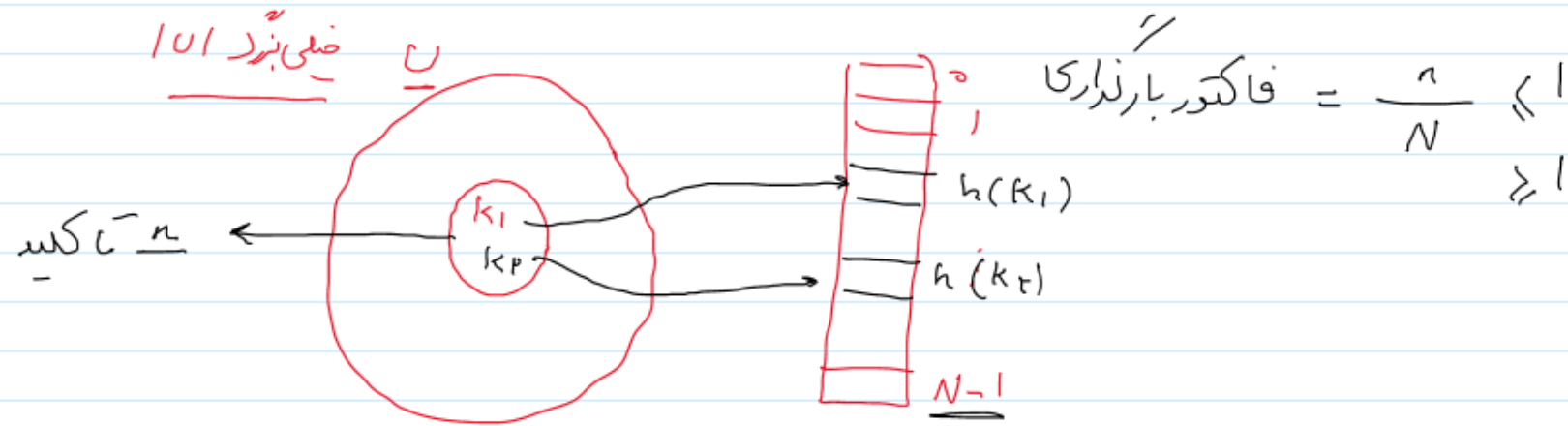
هر جدول در هم سازی  $2^p$  المان دارد:

$$h(n) = x \text{ بر } N$$

$$1- \text{تابع در هم سازی } h: U \rightarrow [0, 1, \dots, N-1]$$

$$2- \text{آرایه: } A = [0, 1, \dots, N-1]$$

ایده اصلی در هم سازی این است که برای درج کلید  $x$ ، آن را در  $A[h(n)]$  ذخیره کنیم



$$\left. \begin{array}{l} U : \text{همه اعداد ۱ رقیبی} \\ n : \omega \text{ کسب} \\ h : \text{عقیم آخر} \\ A : \dots \end{array} \right\} \alpha = \frac{\omega}{1 \dots}$$

collision تصادم : دو کلید با مقدار متفاوت به یک خانه map شوند  $h(x) = h(y)$   $x \neq y$

چالش : چگونه تصادم را حل کنیم ؟

$\alpha$  بر  $N$  گس پذیر باشد !  
 اولاً : بسش گیری همه از درم (۱) تابع هس خود را مناسب انتخاب کنیم

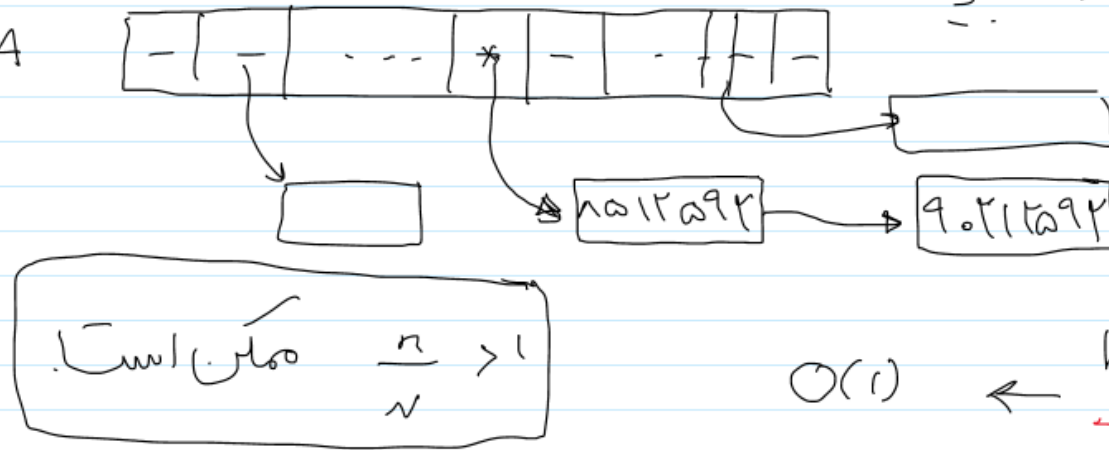
شماره دانشجویی : انتخاب رقم  $X$   $h(x) = (ax + b) \% N$   
 $\downarrow$   
 $(a \% N) = 1$

\* فرض در ادامه درس : تابع هاش ایده آل است : ۱- عضو تصادفی از  $U$  انتخاب کنیم،  
 ۲- احتمال برخورد به هر خانه ای  $m$  شود

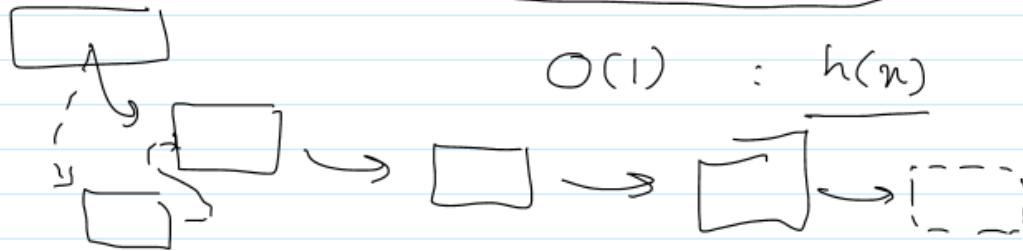
۱- روش اول : روش زنجیره بندی separate chaining

۲ رقم شماره دانشجویی ۲۵۹۲ ۳۱۴۱  
 A : آرایه ای از اشاره لرها

→ ۱۵۱۲۵۹۲  
 → ۹۰۲۱۲۵۹۲  
 ۹۸۲۲۳۱۴۱



حاسبه  $h(n)$  ←  $O(1)$



$O(1) : h(n)$

مقدار :  $O(1)$  درج

$O(n)$  حذف

$O(n)$  جستجو

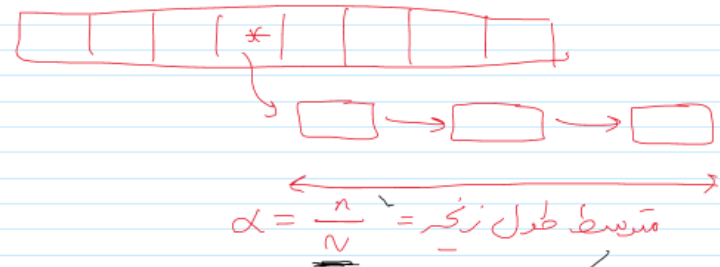
\* حالت متوسط: با فرض ایده آل بودن تابع هاش و این فرض که هر کلید به صورت تصادفی از  $N$  انتخاب شده است.

\* متوسط زمان [حاسب هاش + مقایسه ها]:

حساب موفق:  $O(1) + \frac{\alpha}{2}$

حساب ناموفق:  $O(1) + \alpha$

$O(1 + \frac{\alpha}{2})$



\* مشکل: حافظه اضافی. [ما معمولا دوست داریم تمام کلیدها داخل آرایه ذخیره شوند]

Open Addressing: تمام عناصر حتی در صورت تصادم باز در آرایه  $A$  ذخیره می شوند.

روش های مبتنی بر آدرس دهی باز:

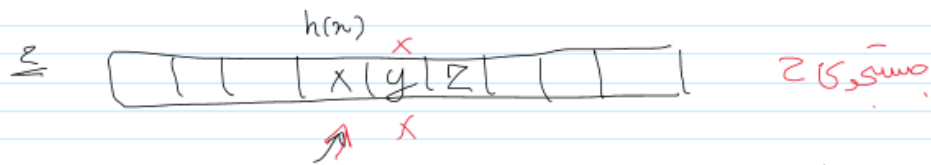
\* linear probing (وارسی خطی): ایده  $h(n)$  را چک کنید، نشد،  $h(n)+1$ ، نشد،  $h(n)+2$  ...

$h(n) = n \times 13$      $19, 31, 32, 59, 44, 22, 41, 19$      $\alpha \leq 1$      $n \leq N$

0	1	2	3	4	5	6	7	8	9	10	11	12
		41		73	19	44	59	32	22	31		

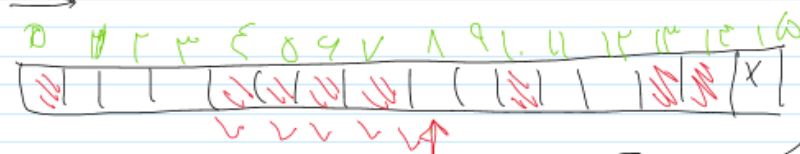
↑   ↑   ↑

حذف:  $\alpha$  : اول جستجو - جستجو:  $\alpha \leftarrow h(n)$  راضی نمی: یا پیدا شود یا به خانه خالی برسم:  $Q(n)$



deleted

مسئله مهم روش واریسی طی:  $N > n$  (primary clustering)

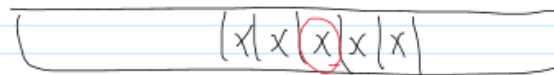


در صورت انتخاب یک عنصر تصادفی از  $N$ ، این عنصر در کدام خانه با احتمال بسته‌ی تعاری لیه؟

در واقع، خانه‌های بعد از خوشه‌های بزرگ، با احتمال بسته‌ی پری شوند.  
لیم تسلیل خوشه‌های بزرگ.

متوسط عملیات جستجو در جستجوی ناموفق:  $\frac{1}{\alpha} \left[ 1 + \frac{1}{1-\alpha^2} \right]$

موفق:  $\frac{1}{2} \left[ 1 + \frac{1}{1-\alpha} \right]$



حذف:

deleted  $\rightarrow$  lazy  
-1  
0