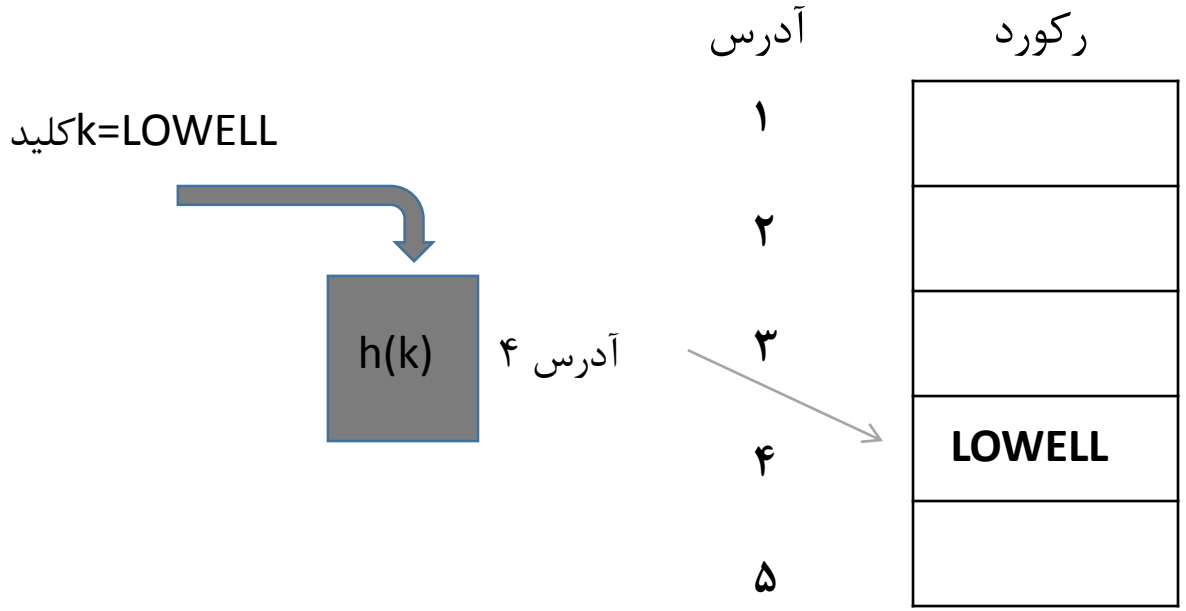
درهم سازی

سید مهدی وحیدی پور

ارایه سوم: درهم‌سازی، روشها و کاربردها

درهم سازی

- درهم سازی تابع h(k) است که کلید k را به یک آدرس انتقال می دهد. از این آدرس به عنوان مبنایی برای ذخیره و بازیابی رکوردها استفاده می‌شود.



به آدرس ۴ آدرس خانگی گفته می شود.

تفاوت درهم سازی و اندیس سازی

- آدرس‌های به دست آمده از درهم سازی تصادفی هستند.
- با درهم سازی دو کلید مختف ممکن است به یک آدرس انتقال داده شوند.که در آن صورت برخورد(Collision) رخ می‌دهد.

چندین روش مختلف برای کاهش تعداد برخوردها

➢**پراکنده کردن رکوردها**

بتوانیم الگوریتمی پیدا کنیم که رکوردها را به صورت تصادفی و پراکنده بین آدرس ها توزیع کند که تعداد زیادی ازرکوردها در اطراف یک آدرس مشخص تجمع نکند.

➢**استفاده از حافظه اضافی**

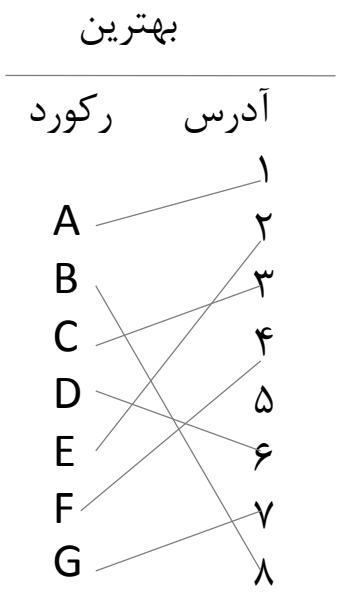
در مقابل تعداد رکورد از حافظه بیشتری کمک می گیریم که پیدا کردن الگوریتم در هم سازی ساده‌تر باشد.

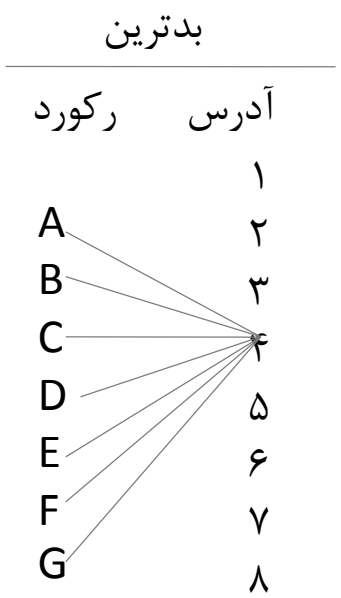
➢**قرار دادن بیشتر از یک رکورد دریک آدرس**

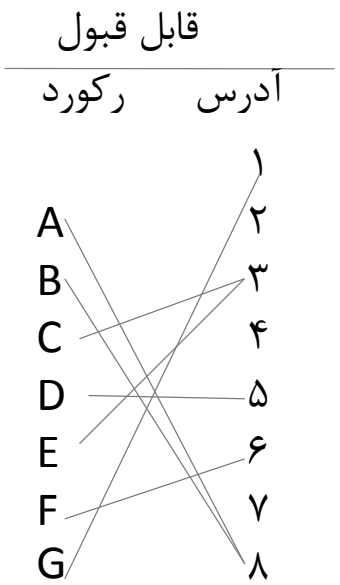
فایل به اندازه کافی بزرگ باشد تا چندین رکورد را در خود ذخیره کند.

به آدرس هایی که چندین رکورد را نگهداری می کنند باکت گفته می شود.

توزیع رکوردها در بین آدرس‌ها







روش های درهم سازی

- **جستجو در کلید برای یافتن یک الگو**: اگر بعضی از قسمت‌های یک کلید، یک الگو واقعی قابل استفاده را نشان دهد. می‌توان از تابع درهم‌سازی استفاده کرد که آن بخش از کلید را استخراج کند.

- **تا کردن قسمت هایی از کلید**: عددهایی را از قسمت‌هایی از کلید استخراج می‌کند و سپس به آنها را به هم اضافه می‌کند. این روش الگوهای اولیه را خراب می‌کند.

- **تقسیم یک کلید بر یک عدد**: اندازه آدرس را بر یک عدد تقسیم می‌کنیم و از باقیمانده‌ی آن استفاده می‌کنیم.

- **مجذور کردن کلید و گرفتن عدد میانی**

- **تبدیل مبنا**

پیش بینی توزیع رکوردها

N تعداد ادرس‌های موجود
R تعداد رکوردهایی که باید ذخیره شوند.
X تعداد رکوردهایی که به یک ادرس مشخص اختصاص داده شده‌اند.

توزیع پواسون

$$P(x)=\frac{(r/n)^x e^{-(r/n)}}{X!}$$

احتمال اینکه هیچ کلیدی در یک ادرس مشخص درهم سازی نشود:

$$P(0)=\frac{1^0\ e^{-1}}{0!}=۰.۳۶۸$$

احتمال اینکه دقیقا یک،دو یا سه کلید در یک ادرس مشخص در هم سازی شوند:

$$P(1)=\frac{1^1\ e^{-1}}{1!}=۰.۳۶۸$$

$$P(2)=\frac{1^2\ e^{-1}}{2!}=۰.۱۸۴$$

$$P(3)=\frac{1^0 3\ e^{-1}}{3!}=۰.۰۶۱$$

دانسیته فشردگی

$$\text{دانسیته فشردگی} = R/N = \frac{\text{تعداد رکورد ها}}{\text{تعداد فضا}}$$

اندازه فضای استفاده شده در فایل را نشان می‌دهد و تنها راه تخمین کارایی محیط درهم‌سازی است.

دانسیته فشردگی– ادامه

اگر دانسیته فشردگی ۵۰ درصد باشد وهر ادرس تنها قادر باشد یک رکورد را در خود ذخیره کند انتظار داریم که ۲۱ درصد تعداد کل رکورد ها در جایی غیر از ادرس خانگی خودشان ذخیره شوند.

اثر دانسیته فشردگی نسبت به رکورد هایی که در ادرس خانگی خود ذخیره نشده اند:

دانسیته فشردگی(درصد)	مترادف ها به صورت درصدی از رکوردها
۱۰	۴.۸
۳۰	۱۳.۶
۵۰	۲۱.۴
۷۰	۲۸.۱
۸۰	۳۱.۲
۹۰	۳۴.۱
۱۰۰	۳۶.۸

از تابع پواسون و دانسیته نتیجه می‌شود که برخورد اجتناب ناپذیر است در نتیجه باید به روش‌های رفع برخورد بیاندیشیم.

روش های رفع برخورد

۱. online(کلید موجود نیست)
- ادرس باز
  - خطی
  - درجه دوم
  - دوگانه
  - دامنه محدود
- زنجیره‌ای

۲.Offline(کلید موجود ۱ست)
- perfect(بدون تصادم)
- minimal perfect
- preserving order MPHF

- روش‌های کاهش تعداد برخوردها
- توزیع رکورد ها بین ادرس ها
- انواع روش‌های درهم‌سازی
- توزیع پواسون و دانسیته فشردگی

- ذخیره کردن بیشتر از یک رکورد در هر ادرس(باکت)
- پیوند با ناحیه سرریز
- جدول های پراکندگی:اندیس سازی

درهم سازی آدرس دهی باز
درهم‌سازی خطی

فرمول کلی

$$H_i(k,i) = (h(k) + c*i) \bmod m \ , \ i = 0,1,\ldots$$

k مقدار کلید
h(k) یک تابع در هم ساز ساده
i نشان دهنده تعداد تکرار تولید آدرس برای کلید k

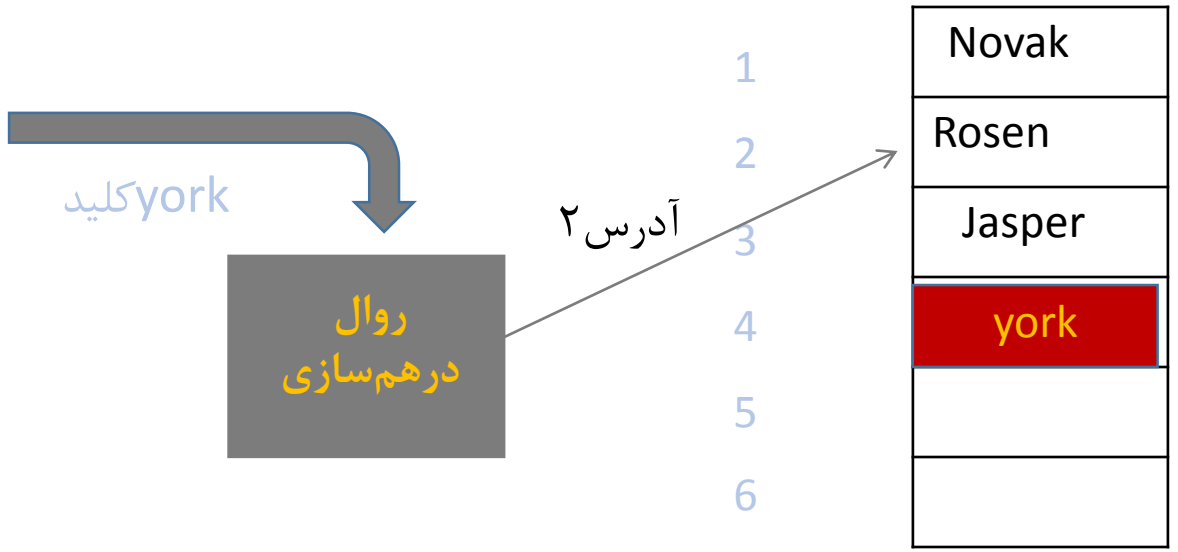
مشکلات:

۱_خوشه بندی اولیه: به ازای مقدار ثابت c وتمام مقادیر اولیه برای تابع h(k,i)=h(k)یک دنباله ادرس مشخص بر اساس میزان جابجایی cتولید خواهد نمود.

۲_خوشه بندی ثانویه: به ازای دو کلید مختلف k_1و k_2 رابطه $h(k_1)=h(k_2)$ برقرار باشد در این صورت تمام ادرس‌های دو دنباله یکسان خواهد بود.

درهم سازی آدرس دهی باز
روش سرریز فزاینده

- درهم سازی خطی با c=1
- اگر به یک رکورد آدرسی تخصیص داده شود که قبلا توسط رکوردی اشغال شده است آدرس‌های بعدی مورد جستجو قرار می گیرند تا اولین آدرس خالی پیدا شود.



❑ هنگام جستجو کلید york جستجو از آدرس ۲ شروع می‌شود زیرا york هنوز در آدرس ۲درهم‌سازی می‌شود.

روش‌های مختلف جانشین‌سازی

- **FCFS**:کلید قدیمی در سلول نگه داشته میشود و کلید جدید سلول های بعدی را بررسی می‌کند.
- **LCFS**:کلید جدید وارد سلول میشود و کلید قدیمی سلول های دیگر را بررسی می‌کند.
- **Robin hood**:در بین دو کلید کلیدی که بزرگترین تغییر مکان را دارد در سلول نگه داشته می‌شود و کلید دیگر سلول‌های دیگر را بررسی میکند.

درهم سازی آدرس دهی باز
درهم‌سازی دوگانه

فرمول کلی

$$HD(k,i) = (g(k) + ih(k)) \bmod m$$

k مقدار کلید

h(k) و g(k) توابع درهم ساز ساده

i نشان دهنده تعداد تکرار تولید آدرس برای کلید k

- **رکورد های سرریز در آدرس دورتری از آدرس خانگی قرار دارند.**
- **هنگام ایجاد برخورد تابع در هم سازی دیگری روی کلید عمل میکند.**

مشکلات:

- ❑ محلی بودن را نقض می‌کند
- ❑ تولید آدرس‌های یکسان و تکراری در دنباله‌های ادرس.

درهم سازی آدرس دهی باز

درهم‌سازی دامنه محدود

به منظور استفاده از روش دامنه محدود، روند زیر برای بدست آوردن آدرس کلید K در فضایی با $m= p^n$ مکان اجرا می شود.
P عدد اول و $n\geq 1$.

- به ازای هر کلید k دو عدد a,b را محاسبه کنید:

$a= k \bmod m, \qquad b = k \bmod (m-2)$

- واضح است که روابط $0\leq a\leq m-1$, $0\leq b\leq m-2$ برقرار می شود.
- مجموعه S را یک مجموعه دامنه محدود $GF(P^n)$می‌نامند. هر چند جمله‌ای از محدوده$GF(P^n)$ را با n عدد ضریب ان نشان میدهند.$a= (a_{n-1}, a_{n-2},..., a_0)_P$ لذا یک عدد Nرقمی در مبنای P نشان دهنده یک چند جمله ای در محدوده $GF(P^N)$است.
- اعداد a وb درمبنای p بر اساس محدوده‌ی $GF(p^n)$ نشان دهنده‌ی دو چندجمله‌ای به نامهای $g_k (X)$ و $h_k (X)$ می‌باشند.

$a= (a_{n-1}, a_{n-2},..., a_0)_P$

$b= (b_{n-1}, b_{n-2},..., b_0)_P$

$g_k (X)= a_{n-1}X^{n-1} + a_{n-2}X^{n-2} + ... + a_0$

$h_k (X)= b_{n-1}X^{n-1} + b_{n-2}X^{n-2} + ... + b_0$

درهم سازی آدرس دهی باز

درهم‌سازی دامنه محدود

- شماره دنباله آدرس ها، که نشان دهنده ی چندمین آدرس تولیدی است، را نیز در مبنای p نمایش داده و چندجمله ای $f_i(x)$ در فضای $GF(p^n)$ تولید می شود.

$$i= (i_{n-1}, i_{n-2},..., i_0)_P$$

$$f_i (X)= i_{n-1}X^{n-1} + i_{n-2}X^{n-2} + ... + i_0$$

- حال بر اساس قوانین حاکم بر $GF(p^n)$ و با در نظر گرفتن $f_\lambda(X)$ به عنوان چند جمله ای ثابت در فضای $GF(p^n)$ رابطه زیر نشان دهنده تابع در هم ساز جدید است.

$$H_f(k,i) = (g_k(X) + f_\lambda(X)\, f_i(X)\, h_k(X)) \bmod t(x)$$

- $H_f(k,i)$ چندجمله ای از فضای $GF(p^n)$ می باشد که ضرایب آن نشان دهنده عددی است در مبنای p که همان آدرس تولید شده تابع درهم‌ساز است.
- $t(x)$ چندجمله‌ای تجزیه ناپذیر در مبنای p مانند

چندجمله‌ای تجزیه ناپذیر
$x^4+x+1, x^4+x^3+x^2+x+1, x^4+x^3+1$
$x^8+x^7+x^6+x^5+x^4+x^3+x^2+1$

ذخیره‌کردن بیشتر از یک رکورد در یک آدرس: باکت ها

- گاهی ذخیره‌سازی گروهی از رکوردها در بلوک‌ها بهتر از ذخیره‌سازی تک تک آنها است.
- در واقع چند رکورد از آدرس مشترک دارند.

- برای جستو یک رکورد کل باکت در حافظه قرار می‌گیرد و رکوردهای موجود در باکت جستجو می‌شوند.

- حالت سرریز در باکت وجود دارد اما احتمال وقوع آن بسیار کمتر است.

حذف رکورد

حذف فایل پیچیده تر از اضافه کردن آن است،زیرا:

- محل رکورد حذف شده نباید مانع جستجو های بعدی شود.
- امکان استفاده از فضای آزاد شده باید وجود داشته باشد.

راه حل: استفاده از علائم ویژه

Adams
Jones
######
Smith

سرریز فزاینده زنجیره‌ای

- رکورد های مترادف در یک لیست پیوندی قرار می‌گیرند. یعنی هر آدرس خانگی حاوی عددی است که محل رکورد بعدی که در این آدرس خانگی قرار دارد مشخص می‌کند.

طول جستجو	آدرس مترادف بعدی	داده
1	22	Adams
1	23	Bates
2	25	Cole
2	-1	Dean
1	-1	Evans
3	-1	Flint

آدرس واقعی	آدرس خانگی
۲۰	۲۰
۲۱	۲۱
۲۲	۲۰
۲۳	۲۱
۲۴	۲۰
۲۵	۲۴

متوسط طول جستجو: ۱.۷

مزیت: در هر جستجو فقط رکوردهایی که دارای کلیدهای مترادف هستند دستیابی می‌شوند.

پیوند با ناحیه سرریز

تمام رکورد های سرریز را به یک ناحیه سرریز منتقل می‌کنیم.

مجموعه آدرس های خانگی **ناحیه داده اصلی** و مجموعه آدرس های سرریز **ناحیه سرریز** نام دارد.

ناحیه سرریز

Cole	2
Dean	-1
Flint	-1

0	
1	
2	
3	

آدرس خانگی	ناحیه داده اصلی	
۲۰	Ádams	0
۲۱	Bates	1
۲۲		
۲۳		
۲۴	Evans	-1

جدول های پراکندگی:اندیس سازی

- فایل در هم سازی حاوی هیچ رکوردی نیست و فقط حاوی اشاره گرهایی به رکوردهاست.
- این فایل مانند یک اندیس است که توسط درهم سازی جستجو می‌شود.