



یادآوری جلسه پانزدهم Map و جدول درهم‌سازی

در جلسه‌ی قبل ابتدا داده‌ساختار Map را مورد بررسی قرار دادیم و سپس در مورد جدول درهم‌سازی صحبت کردیم. ابتدا به تعریف داده ساختار Map می‌پردازیم:

این داده‌ساختار تعدادی دوتایی را به صورت (key,value) (با کلیدهای یکتا) نگه می‌دارد و می‌تواند عملیات‌های زیر را انجام دهد:

۱. $insert(key,value)$: درج دوتایی داده شده در داده ساختار

۲. $search(key)$: جستجوی کلید و برگرداندن مقدار آن

۳. $delete(key)$: حذف دوتایی با کلید داده شده

این داده ساختار را به روش‌های زیر می‌توان پیاده سازی کرد:

۱. آرایه‌ی نامرتب: یک آرایه برای کلیدها و یک آرایه برای مقادیر در نظر می‌گیریم. برای درج ، دوتایی مورد نظر را در اولین خانه‌ی خالی آرایه ذخیره می‌کنیم. درج از مرتبه $O(1)$ و جست‌وجو و حذف از مرتبه $O(n)$ می‌باشد.

۲. آرایه‌ی مرتب: مشابه آرایه‌ی نامرتب است. با این تفاوت که دوتایی‌ها را به‌گونه‌ای درج می‌کنیم که آرایه براساس کلیدها مرتب باشد. درج و حذف در $O(n)$ و جستجو در $O(\log n)$ انجام می‌شود.

۳. ددج متوازن: هر گره درخت شامل یک دوتایی است و درخت براساس کلیدها مرتب شده‌است. می‌دانیم عملیات‌های درج، جستجو و حذف در ددج متوازن از مرتبه‌ی زمانی $O(\log n)$ هستند.

حال جدول درهم‌سازی را مورد بررسی قرار می‌دهیم. هر جدول درهم‌سازی شامل:

۱. تابع درهم‌سازی: $h : U \rightarrow \{0, 1, \dots, N-1\}$

۲. آرایه A : اندازه‌ی $N = A$

مجموعه‌ی تمام کلیدهای ممکن را با U و تعداد دوتایی‌ها را با n نشان می‌دهیم.

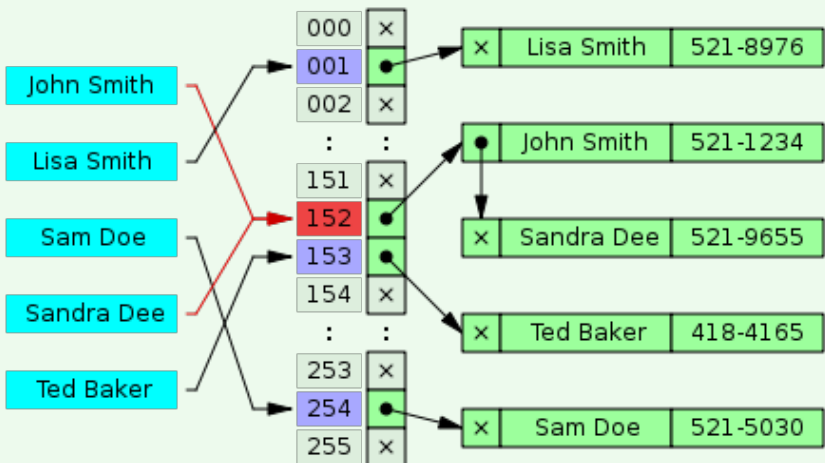
ایده کلی برای درج x در جدول درهم‌سازی این است که مقدار value را در $A[h(x)]$ ذخیره می‌کنیم.

برابری مقدار تابع درهم‌سازی برای دو کلید متفاوت را **تصادم** می‌گوییم. برای رفع تصادم یکی از روش‌های زیر را در پیش می‌گیریم:

- پیشگیری از وقوع تصادم: تابع درهم‌سازی مناسبی انتخاب کنیم.

تابع درهم‌سازی ایده‌آل ، تابع درهم‌سازی است که اگر یک کلید تصادفی از U را انتخاب کنیم، احتمال Map شدن آن به هر یک از خانه‌های آرایه یکسان باشد.

- رفع تصادم: در جلسه قبل رفع تصادم به کمک روش زنجیره‌بندی را بررسی کردیم. در این روش هر خانه از آرایه‌ی A یک زنجیره از دوتایی‌هایی است که کلید دوتایی‌ها به آن خانه آرایه Map شده‌است.



شکل ۱: در این دیکشنری هر کلید یک اسم است که به عددی ۷رقمی مربوط می‌شود. N برابر با ۲۵۶ است بنابراین تابع درهم‌سازی هر اسم را به یکی از اعداد ۰ تا ۲۵۵ تصویر می‌کند. اگر مانند خانه‌ی ۱۵۲ آرایه، دو اسم به یک خانه تصویر شوند، به صورت زنجیره‌ای به هم متصل می‌شوند.

فرض می‌کنیم محاسبه‌ی $h(x)$ در $O(1)$ انجام می‌شود، در این صورت در بدترین حالت خواهیم داشت:

- درج: $h(x)$ را محاسبه کرده و دوتایی را به ابتدای زنجیره اضافه می‌کنیم. $O(1)$
- جستجو: $h(x)$ را محاسبه کرده و جستجو را تا انتهای زنجیره ادامه می‌دهیم. $O(n)$
- حذف: $h(x)$ را محاسبه کرده و جستجو را تا انتهای زنجیره ادامه می‌دهیم و سپس دوتایی مورد نظر را حذف می‌کنیم. $O(n)$

در حالت متوسط خواهیم داشت:

فرض می‌کنیم که کلیدها به‌صورت تصادفی از U انتخاب شوند و تابع $h(x)$ نیز ایده‌آل باشد. در این صورت n درج داریم و اندازه‌ی آرایه برابر با N است. بنابراین طول هر زنجیره به طور متوسط برابر با n/N است که ضریب بازگذاری می‌نامیم و با α نشان می‌دهیم.

- درج: $O(1)$

- جست‌وجو:

– ناموفق: $h(x)$ را محاسبه کرده و تا انتهای زنجیره پیش می‌رویم. $O(1 + \alpha)$

– موفق: $h(x)$ را محاسبه کرده و به طور متوسط تا میانه زنجیره پیش می‌رویم. $O(1 + \frac{\alpha}{2})$

- حذف: ابتدا جستجو را انجام داده و عنصر با کلید مورد نظر را می‌یابیم، سپس آن را حذف می‌کنیم.

پرسش: فرض کنید یک کلاس داریم که ۷ دانشجو با شماره دانشجویی‌های

۴۸۵۳۹۱۲۰, ۴۸۳۹۱۰۲۸, ۸۵۱۴۹۲۳, ۱۲۰۴۹۳۸۵, ۹۰۲۱۰۵۲۶, ۹۹۶۵۹۴۹۹, ۱۰۲۹۹۰۰۱ و

یک آرایه A با اندازه ۹ داریم. یک تابع درهم‌سازی پیشنهاد دهید که افراد را بدون تصادم در آرایه قرار

دهد.

پاسخ‌های خود را به [این لینک](#) ارسال کنید.

