



در جلسه قبل، بحث راجع به درهم‌سازی را آغاز کردیم. همانطور که اشاره شد، فرض کنید نیاز به داده ساختاری داریم که این اعمال را انجام دهد: $insert(key, value)$ و $delete(key)$ و $search(key)$. دقت کنید که در اینجا فرض شده است که کلیدها یکتا است (هیچ دو مقداری با کلید یکسان وجود ندارند). روش‌های ساده پیاده‌سازی این ADT مانند پیاده‌سازی با استفاده از آرایه، آرایه مرتب و درخت جستجوی دودویی متوازن را بررسی کردیم. با این حال، با توجه به کاربردی که این داده ساختار دارد، نیاز به این است که این اعمال را در زمان سریع‌تری انجام دهیم.

حال فرض کنید که میدانیم کلیدها عدد و در بازه $[0, 9999]$ است. در این صورت، یک روش پیشنهادی استفاده از یک آرایه A با اندازه 10000 است. به این صورت که اگر دوتایی (x, y) قرار است درج شود، در خانه $A[x]$ مقدار y قرار داده می‌شود (فرض کنید یک مقدار $null$ در نظر گرفته‌ایم که نشان دهد خانه خالی است و در ابتدا همه خانه‌ها $null$ است). به این روش، آدرس دهی مستقیم می‌گویند. بدیهی است که در این روش می‌توان تمام عملیات‌ها را در زمان $O(1)$ انجام داد.

اما این روش یک عیب بزرگ دارد: ممکن است تعداد کلیدها متناسب با اندازه بازه کلیدها نباشد. به عنوان مثال، فرض کنید قرار است دوتایی‌های ما به صورت شماره دانشجویی (کلید) و نام و نام خانوادگی (مقدار) افراد کلاس خودمان باشد. برای استفاده از روش آدرس دهی مستقیم، نیاز به یک آرایه به اندازه 100000000 است و نهایتاً 50 خانه از این آرایه قرار است پر شود و این هدر رفت حافظه بالایی دارد. در این مرحله است که استفاده از تابع درهم‌سازی معنی پیدا می‌کند.

۱ جدول درهم‌سازی

فرض کنید مجموعه U مجموعه همه کلیدهای ما است و می‌دانیم که اندازه U بسیار بزرگ است. همچنین می‌دانیم که تعداد ورودی‌های ما حداکثر n کلید دارد. یک جدول درهم‌سازی شامل:

۱. یک تابع درهم‌ساز به صورت

$$h : U \rightarrow [0 \dots N - 1]$$

۲. یک آرایه با اندازه N

است. ایده اصلی درهم‌سازی، این است که مقدار مربوط به کلید ورودی x را در خانه $h(x)$ ذخیره کنیم.

همچنین ضریب n/N به عنوان فاکتور بارگزاری یک جدول درهم‌سازی تعریف می‌شود. به طور معمول ما انتظار داریم که مقدار n و N با هم فاصله زیادی نداشته باشند و حافظه مورد استفاده ما متناسب با تعداد کلیدهای ورودی باشد. دقت کنید که فرض بر این است که $n, N \gg |U|$ است. بزرگترین چالشی که در بحث مربوط به درهم‌سازی با آن مواجه هستیم، مساله تصادم است. یعنی این که در صورتی که دو کلید ورودی به یک خانه هش شدند، چه کنیم؟

پرسش: در این لینک می‌توانید لیستی از شماره دانشجویی همه دانشجویان کلاس را می‌توانید مشاهده نمایید. دو تابع درهم‌سازی برای حالت‌های زیر پیشنهاد دهید و اندازه آرایه درهم‌سازی را برای آنها مشخص کنید. طبعاً هر چقدر اندازه آرایه کوچکتر باشد مطلوب‌تر است.

• می‌خواهیم هیچ تصادمی رخ ندهد.

• کمی تصادم در هر خانه رخ دهد ایرادی ندارد.

پاسخ‌های خود را به این لینک ارسال کنید.

