ساختمان داده‌ها و الگوریتم‌ها

نیم‌سال اول ۱۴۰۰ـ۱۴۰۱

مدرس: مسعود صدیقین

دانشکده‌ی مهندسی کامپیوتر

تمرین سری نهم

مسئله‌ی ۱. هش جهانی (هش جهانی)

فرض کنید که مجموعه کلیدهای ما به صورت بردارهای $[\,x_۱, x_۲, \cdots, x_k\,]$ باشند، با این فرض که هر کدام از x_i ها می‌توانند از بازه $\{۰, ۱, \cdots, N-۱\}$ انتخاب شده باشند که N یک مقدار اول است. نشان دهید که خانواده زیر برای این تابع یک خانواده جهانی است:

$$H \;=\; \{h|\, h(x) = (\, r_۱x_۱ + r_۲x_۲ + \cdots + r_kx_k)\ \%\ N\}$$

که در آن تمام r_iها از بازه $\{۰, ۱, \cdots, N-۱\}$ انتخاب شده‌اند.

مسئله‌ی ۲*. خانواده دوست‌داشتنی (هش جهانی)

اگر p یک عدد اول بزرگ باشد و تعریف کنیم $\mathbb{Z}_p = \{۰, ۱, \cdots, p-۱\}$ ، فرض کنید مجموعه مرجع کلیدهای ما برابر $U = \mathbb{Z}_p$ باشد. اندازه جدول هش را برابر با عدد طبیعی m تعریف کنیم که $m < p$ است. به ازای هر $a \in \mathbb{Z}_p - \{۰\}$ و $b \in \mathbb{Z}_p$ تابع هش $h_{a,b}$ را برای کلید $k \in U$ به صورت زیر تعریف می‌کنیم:

$$h_{a,b}(k) = (\,(\,ak + b)\ mod\, p)\ mod\, m$$

اگر $H_{p,m}$ مجموعه تمام توابع هش به صورت $h_{a,b}$ باشد، ثابت کنید $H_{p,m}$ یک خانواده توابع هش جهانی است.

مسئله‌ی ۳. بدترین زنجیر (زنجیره‌سازی)

فرض کنید تعداد کلیدهایی که می‌خواهیم ذخیره کنیم n باشد و از طرفی اندازه جدول درهم‌سازی m باشد که برای رفع تصادم‌های آن از روش زنجیره‌سازی استفاده می‌کنیم. اگر کلیدها همگی متعلق به مجموعه U باشند و بدانیم که $|U| < nm$ است، ثابت کنید لزوما زیرمحموعه‌ای از U وجود دارد که همگی آن‌ها به یک جایگاه map می‌شوند. و توضیح دهید که این یعنی بدترین مرتبه زمانی جست و جو از مرتبه $\Theta(n)$ است.

مسئله‌ی ۴*. عجیبه ولی واقعیت داره (وارسی مربعی)

یک جدول هش با اندازه m را در نظر بگیرید به طوری که m توانی از ۲ باشد. همچنین تابع هش h را درنظر بگیرید که هر کلید را به یک جایگاه عضو $\{۰, ۱, \ldots, m\}$ مپ می‌کند. برای یافتن عنصر k الگوریتم زیر باید طی شود:

۱. $j = h(k)$ را محاسبه کن و قرار بده $i = ۰$

۲. در جایگاه j به دنبال k بگرد، اگر آن را یافتی یا به جایگاه خالی رسیدی الگوریتم را متوقف کن

۳. قرار بده $i = i + 1$. اگر i برابر m شد، الگوریتم را متوقف کن. در غیر این صورت قرار بده $j = (i + j) \, mod \, m$

الف) نشان دهید که این شیوه جست‌وجو نوعی وارسی مربعی است و مقادیر c_1 و c_2 را برای آن تعیین کنید.
ب) ثابت کنید که این الگوریتم در بدترین حالت تمامی جایگاه‌ها را جست‌وجو می‌کند.

مسئله‌ی ۵*. ب.م.م موثر (درهم‌سازی دوگانه)

در یک جدول درهم‌سازی برای رفع تصادمات از درهم‌سازی دوگانه استفاده می‌کنیم، یعنی $h(k, i) = (h_1(k) + ih_2(k)) \, mod \, m$. نشان دهید اگر برای مقدار دلخواه k، بزرگترین مقسوم‌علیه مشترک m و $h_2(k)$ برابر $d \geqslant 1$ باشد، آنگاه هر جست‌وجوی ناموفق برای k، قبل از برگشتن به خانه $h_1(k)$، تعداد $\frac{1}{d}$ خانه از جدول درهم‌سازی را بررسی می‌کند. بنابراین اگر m و $h_2(k)$ نسبت به هم اول باشند(در آن صورت $d = 1$ می‌شود)، جست‌وجو ممکن است کل جدول درهم‌سازی را بررسی کند.

مسئله‌ی ۶*. آش شله قلمکار (تصادم)

یک گراف 2^n راسی داریم که راس‌های آن از ۰ تا $2^n - 1$ شماره گذاری شده اند. تابع درهم ساز زیر را برای هر راس مثل v تعریف می‌کنیم

$$Hash(v) = xor_{u \in N(v)} u$$

یعنی تابع درهم ساز برای هر راس برابر xor بیتی همسایه‌های آن است. منظور از xor بیتی دو عدد این است که در مبنای دو رفم به رفم آن دو عدد را xor کنیم با فرض اینکه $n \geqslant 2$ باشد و گراف کاملا رندوم باشد ثابت کنید احتمال اینکه برای دوتا راس دلخواه که بهم یال ندارند مثل v, u تابع در هم ساز آن‌ها یکسان باشد برابر 2^{-n} است. یعنی

$$P[Hash(v) = Hash(u)] = 2^{-n}$$

(احتمال وجود داشتن هر یال در گراف برابر ۱/۲ است)

مسئله‌ی ۷*. ایزی ایزی (آدرس‌دهی مستقیم)

یک آرایه بیتی (صفر و یکی) m تایی فضای بسیار کمتری از آرایه‌ای از m پوینتر می‌گیرد. در صورتی که عناصری که داریم، داده اضافی همراه خود نداشته باشند و صرفا مقادیری بین ۰ تا $m - 1$ داشته باشند، نشان دهید چگونه روش آدرس‌دهی مستقیم را می‌توان با آرایه بیتی پیاده سازی کرد.

مسئله‌ی ۸*. کوکو سبزی (هش کامل)

یکی دیگر از روش‌های درهم‌سازی استفاده از روش درهم‌سازی کوکو می‌باشد. در این روش دو جدول درهم‌سازی T_1 و T_2 داریم که هرکدام می‌تواند تابع درهم‌سازی مخصوص خود را داشته باشد ($h_1(k)$, $h_2(k)$). برای جست‌وجوی عنصر k تنها لازم است دو خانه‌ی $T_1[h_1(x)]$ و $T_2[h_2(x)]$ را بررسی کنیم و اگر عنصر موردنظر درهیچکدام از این دو خانه نبود این عنصر هیچوقت درج نشده‌است.

برای درج نیز ابتدا این دو خانه را بررسی می‌کنیم. اگر حداقل یکی از آن‌ها خالی بود عنصر را در یکی از آن‌ها درج می‌کنیم و کار تمام است. درغیر این‌صورت عنصر k را در $T_۱[h_۱(k)]$ درج می‌کنیم و عنصری که قبلاً در این خانه بوده را به جدول $T_۲$ منتقل می‌کنیم و این کار را به‌صورت بازگشتی روی این عنصر ادامه می‌دهیم. برای اطلاعات بیشتر درمورد این روش می‌توانید جست‌وجو کنید.

حال فرض کنید دو جدول کوکو داریم که هرکدام ۱۰ خانه دارند. مساله‌ی ۱ را با استفاده از این روش حل کنید (دقت کنید باید دو تابع مناسب نیز پیشنهاد دهید).

همچنین چه روش‌هایی پیشنهاد می‌کنید که از افتادن در دور بی‌نهایت جلوگیری کند؟

مسئله‌ی ۹*. کاوش زنجیره‌ای (کاوش خطی)

الف) در پیاده‌سازی دو درهم‌سازی، در یکی برای تصادم‌ها از روش زنجیر و در دیگری از کاوش خطی استفاده شده است. فرض کنید تعداد قابل توجهی تصادم داشته باشیم، کدام یک بهینه‌تر است؟
ب) در قسمت الف فرض کنید به جای کاوش خطی از کاوش درجهٔ ۲ استفاده کنیم. اکنون کدام بهتر است؟