طراحی الگوریتم‌ها

نیم‌سال دوم ۰۰ـ۹۹

مدرس: مسعود صدیقین

دانشکده‌ی مهندسی کامپیوتر

تمرین دوم

مسئله‌ی ۱ *. کسرهای مصری (الگوریتم‌های حریصانه)

فرض کنید که $\frac{x}{y}$ یک عدد کسری حقیقی بین ۰ و ۱ است. به نمایش زیر برای $\frac{x}{y}$ یک نمایش مصری گفته می‌شود.

$$\frac{x}{y} = \frac{۱}{k_۱} + \frac{۱}{k_۲} + \frac{۱}{k_۳} + ...$$

در این نمایش تمامی k ها عدد طبیعی و یکتا هستند. برای مثال می‌توان عدد $\frac{۳}{۷}$ به شکل زیر نوشت.

$$\frac{۳}{۷} = \frac{۱}{۳} + \frac{۱}{۱۱} + \frac{۱}{۲۳۱}$$

یک الگوریتم حریصانه برای پیدا کردن نمایش مصری هر $\frac{x}{y}$ ارائه دهید.

مسئله‌ی ۲ *. حافظه ترتیبی (الگوریتم‌های حریصانه)

در این سؤال ما می‌خواهیم یک حافظه ترتیبی [۱] برای n تا المان ذخیره‌سازی طراحی کنیم. حافظه ترتیبی است یعنی عناصر حافظه ترتیب دارند و برای دسترسی به عنصر iام در حافظه باید به ترتیب تمامی عناصر ۱ تا i حافظه را ببینیم و نهایتا به المان مورد نظر می‌رسیم لذا اگر المانی در جایگاه iام قرار داشته‌باشد، دسترسی به آن i مرحله طول خواهد کشید.

می‌دانیم میزان دسترسی به این المان‌های ذخیره‌سازی یکسان نیست، بعبارتی یک‌سری داده‌ها از یک‌سری دیگر پر کاربردترند و لذا بیشتر سعی می‌کنیم به آن‌ها دسترسی داشته‌باشیم. فرض کنید احتمال دسترسی به iامین داده یا المان ذخیره‌سازی برابر p_i است. همچنین هر داده یک تاخیر [۲] نیز دارد، به ازای داده iام این مقدار را برابر l_i تعریف می‌کنیم. تاخیر یک داده مقدار زمانی هست که طول می‌کشد تا از روی آن داده بگذریم. برای مثال اگر دو داده $l_۱ = ۱۵$ و $l_۲ = ۲۰$ را به همین ترتیب بچینیم، برای دسترسی به داده اول ۱۵ واحد زمان مصرف می‌شود و برای دسترسی به داده دوم $l_۱ + l_۲ = ۱۵ + ۲۰ = ۳۵$ واحد زمان مصرف می‌شود.

فرض کنید p_i و l_i ها را داریم، ترتیبی بهینه برای داده‌ها در حافظه ارائه دهید و اثبات کنید این ترتیب کمترین امیدریاضی مدت زمان دسترسی [۳] به یک خانه از حافظه را می‌دهد.

مسئله‌ی ۳ *. صف (الگوریتم‌های حریصانه)

فرض کنید که n نفر وارد یک بانک با m باجه شده‌اند و کار هر کدام t_i دقیقه طول می‌کشد. به d_i میزان معطلی هر

---

[۱] Sequential

[۲] latency

[۳] امید ریاضی مدت زمان دسترسی به شکل روبرو تعریف می‌شود : $\sum_{i=۱}^{n} p(i) \times l(i)$

فرد می‌گوییم. اگر فردی نفر jام صف یکی از باجه‌ها باشد، آنگاه $d_j = \sum_{k=1}^{j} t_k$. در این رابطه t_k مدت زمان کار افراد درون صف است. یک چینش از این n نفر در صف باجه‌ها ارائه دهید که در نهایت مجموع میزان معطلی افراد را کمینه کند. در نهایت ثابت کنید این چینش همیشه پاسخ بهینه را می‌دهد.

مسئله‌ی ۴*. رشته‌ها (الگوریتم‌های حریصانه)

فرض کنید n رشته داریم که همگی از کاراکترهای a و b تشکیل شده‌اند. می‌خواهیم این رشته‌ها را به ترتیبی به هم بچسبانیم به طوریکه تعداد جفت مکان‌هایی مثل i و j که $i < j$ است و در جایگاه i کاراکتر a ظاهر شده ولی در جایگاه j کاراکتر b ظاهر شده کمینه باشد. برای مثال اگر دو رشته abb و aab داشته‌باشیم. می‌توانیم $abbaab$ یا $aababb$ را بسازیم در اولی تعداد این جفت جایگاه‌ها ۵ تا و در دومی ۸ تاست بنابراین حالت اول را ترجیح می‌دهیم. فرض کنید جمع طول رشته‌ها برابر S است. الگوریتمی از پیچیدگی زمانی $O(S + n\log n)$ برای پیدا کردن این ترتیب ارائه کنید.

مسئله‌ی ۵. ماتریس صفر و یک (الگوریتم‌های حریصانه)

فرض کنید درایه‌های یک ماتریس $n \times n$ اعداد صفر و یک هستند. به ازای هر سطر و ستون تعداد یک‌های آن داده‌شده‌است. ماتریسی بسازید که تعداد یک‌ها و صفرهای آن با آنچه داده‌شده‌است مطابقت کند.

مسئله‌ی ۶*. ترکاندن بادکنک‌ها (الگوریتم‌های حریصانه)

تعدادی دایره با شعاع‌های نه لزوما یکسان روی صفحه قرار گرفته‌اند، به گونه‌ای که همه بالای محور x هستند و مساحت هیچ دو دایره‌ای اشتراک ندارد. شما روی مبدا مختصات ایستاده‌اید و در هر مرحله می‌توانید یک تیر پرتاب کنید. تیری که پرتاب می‌کنید روی یک خط راست حرکت می‌کند و به هر دایره‌ای که برخورد می‌کند و آن را می‌ترکاند. به بیان دیگر، مسیر حرکت هر تیر یک نیم‌خط است که از مبدا مختصات آغاز می‌شود و هر دایره‌ای که با این نیم‌خط برخورد داشته باشد یا بر آن مماس گردد، می‌ترکد. الگوریتمی با زمان $O(n\log n)$ ارائه دهید که با کمترین تعداد تیر، همه‌ی دایره‌ها را بترکاند. درستی الگوریتم خود را اثبات نمایید. .

مسئله‌ی ۷. خرد کردن پول (الگوریتم‌های حریصانه)

- الف) در شهری ، k نوع سکه با ارزش های $a^0, a^1, ..., a^{k-1}$ وجود دارد. الگوریتمی حریصانه برای خرد کردن پول در این شهر ارائه دهید تا برای این‌کار از کمترین تعداد سکه استفاده شود.
- ب) در حالت کلی الگوریتم حریصانه‌ای که در قسمت قبل به کار می‌بریم به بهترین جواب نمی‌رسد و مثالی برای نقض الگوریتم خود در قسمت قبل بیاورید.

مسئله‌ی ۸*. دنبال‌ی مثبت (الگوریتم‌های حریصانه)

فرض کنید یک آرایه $A[1 \ldots n]$ از اعداد صحیح داده‌شده‌است. یک زیردنباله‌ی پیوسته $A[i \ldots j]$ یک بازه‌ی مثبت گفته می‌شود اگر جمع اعضای $A[i]$ تا $A[j]$ بزرگ‌تر از صفر باشد. می‌خواهیم کمترین تعداد بازه‌های مثبت که

همه‌ی اعداد مثبت را پوشش دهد را پیدا کنیم. الگوریتمی حریصانه ارائه دهید و درستی آن را اثبات کنید و در نهایت زمان اجرای الگوریتم خود را محاسبه کنید.

مسئله‌ی ۹. کمبود پارکینگ (الگوریتم‌های حریصانه)

در یک پارکینگ تعدادی ماشین و تعدادی محل برای پارک ماشین وجود دارند. در واقع می‌توانید یک پارکینگ را به شکل آرایه‌ای در نظر بگیرید که در هر خانه از آن C به نشانه‌ی ماشین یا P به نشانه‌ی محل پارک قرار دارد. هر ماشین فقط می‌تواند در یکی از محل‌های پارک ماشین، پارک کند ولی رانندگان ترجیح می‌دهند که ماشین خود را حداکثر در شعاع k از محل فعلیش پارک کنند و در صورتی که فاصله‌ی بین ماشین و محل پارک بیشتر از k باشد از پارک کردن ماشین خود منصرف می‌شوند.
الگوریتمی حریصانه برای یافتن بیشترین تعداد ماشینی که در این پارکینگ می‌توان پارک کرد ارائه دهید.
برای مثال خروجی برای آرایه‌ی زیر و شعاع ۲ برابر ۳ خواهد بود.

$$\{P, P, C, C, P, C\}$$