الگوریتم های حریصانه

۱ مقدمه ای بر روش حریصانه

در این بخش و بخشهای آینده به معرفی چند تکنیک اساسی برای طراحی الگوریتمها می‌پردازیم. عموما مسائلی را که مورد بررسی قرار می‌دهیم از جنس مسائل بهینه سازی گسسته هستند. به این معنی که برای حل یک مسئله، در یک مجموعه متناهی از جوابهای ممکن، دنبال جوابی می‌گردیم که با توجه به یک معیار خاص بهترین جواب باشد. البته جواب مسئله ممکن است منحصر بفرد نباشد. در هر صورت منظور از حل مسئله، طراحی الگوریتمی است که جواب درست و بهینه را به سرعت پیدا کند.

واقعیت این است که روش حریصانه در طراحی الگوریتمها تعریف روشن و مشخصی ندارد. عموما الگوریتمهای حریصانه الگوریتمهایی هستند که در قدمهای کوچک و غیر قابل بازگشت و با اتخاذ تصمیمات نزدیک بینانه عمل میکنند. اگر بخواهم مثالی در این مورد بزنم، فرض کنید شما مقداری پول دارید و میخواهید بیشترین استفاده از آن را ببرید. یک راه حریصانه این است که با مقدار پولی که دارید بیشترین تعداد جنس را بخرید و لذا طبیعتا دنبال اجناس ارزان قیمت میگردید. این یک نوع تصمیم گیری نزدیک بینانه و حریصانه است. شاید در بلند مدت این سیاست به سود شما نباشد. احتمال دارد اجناسی که خریده اید کیفیت بالایی نداشته و پس از مدتی کوتاه نیاز به خرید مجدد و تعویض آنها پیدا کنید.

در بعضی از موقعیتها روش حریصانه استراتژی خوبی است و شما را به جواب بهینه میرساند اما در بسیاری از موقعیتها تصمیمات نزدیک بینانه و حریصانه اگر چه به سرعت شما را به یک جواب میرساند، اما لزوما جواب بدست آمده بهینه نیست و ممکن است حتی خیلی بد هم باشد.

در این فصل به بررسی مسائلی میپردازیم که روشهای حریصانه در آنها موفقیت آمیز بوده و جواب بهینه را به ما می دهد. در فصلهای بعدی، سراغ مسائلی میرویم که روشهای حریصانه جوابگو نیست و تکنیکهای دیگری برای حل آنها مورد نیاز است.

۲ مسئله صندوقدار

فرض کنید سکه‌هایی به ارزش ۱ ریال، ۵ ریال، ۱۰ ریال، ۲۵ ریال و ۱۰۰ ریال داریم (توجه کنید در شکل زیر سکه ۲۵ ریالی در واقع ساختگی است و هیچ وقت ضرب نشده!). عدد صحیح و مثبت x داده شده و می‌خواهیم با استفاده از کمترین تعداد سکه ارزشی معادل x ریال را داشته باشیم. به عبارت دیگر x ریال را با استفاده از کمترین تعداد سکه خرد کنیم. یک استراتژی حریصانه برای این مسئله این است که از بزرگترین سکه ممکن شروع کنیم و برای مقدار باقیمانده هم به همین منوال ادامه دهیم. پرسش اینجاست که آیا این استراتژی به کمترین تعداد سکه منتهی می‌شود؟

می‌توان اثبات کرد که استراتژی حریصانه برای نوع سکه‌هایی که گفته شد جواب بهینه را بدست می‌آورد. این ادعا را با استقرا روی مقدار x اثبات می‌کنیم. برای $x \leq 4$ روشن است که حریصانه جواب بهینه را بدست می آورد (پایه استقرا) حال فرض کنید برای $x < n$ حریصانه همواره جواب بهینه را نتیجه می‌دهد (فرض استقرا). برای اثبات برای حالت $x = n$ گزینه‌هایی زیر را بررسی می‌کنیم. قبل از بررسی گزینه ها چند مشاهده انجام می‌دهیم.

- جواب بهینه هیچ گاه بیشتر از ۴ سکه ۱ ریالی استفاده نمی‌کند.
- جواب بهینه هیچ گاه بیشتر از یک سکه ۵ ریالی استفاده نمی‌کند.
- در جواب بهینه، مجموع تعداد سکه‌هایی ۵ ریالی و ۱۰ ریالی حداکثر ۲ است.
- جواب بهینه هیچ گاه بیشتر از ۳ سکه ۲۵ ریالی استفاده نمی‌کند.

حال به بررسی گزینه‌ها می‌پردازیم:

۱. $5 \leq x < 10$:

اگر جواب بهینه از سکه ۵ ریالی استفاده کرده باشد، روشن است که جواب بهینه و حریصانه یکی است. فرض کنید جواب بهینه از سکه ۵ ریالی استفاده نکرده باشد. پس تنها راهش این است که x سکه ۱ ریالی استفاده کند و این با مشاهدات بالا متناقض است.

۲. $10 \leq x < 25$:

اگر جواب بهینه از سکه ۱۰ ریالی استفاده کرده باشد، با فرض استقرا استراتژی حریصانه برای $x - 10$ درست جواب می‌دهد. پس در کل برای مقدار x استراتژی حریصانه درست است. حال فرض کنید جواب بهینه از سکه ۱۰ ریالی استفاده نکرده باشد. پس تنها راهش این است که از سکه های ۵ ریالی و ۱ ریالی استفاده کند. اما با توجه به مشاهدات بالا با استفاده از این سکه‌ها حداکثر ارزش ۹ قابل جمع آوری است. لذا این با $x \geq 10$ متاقض است.

۳. : $25 \leq x < 100$

اگر جواب بهینه از سکه ۲۵ ریالی استفاده کرده باشد، با فرض استقرا استراتژی حریصانه برای $x - 25$ درست جواب می‌دهد. پس در کل برای مقدار x استراتژی حریصانه درست است. حال فرض کنید جواب بهینه از سکه ۲۵ ریالی استفاده نکرده باشد. پس تنها راهش این است که از سکه های ۱۰ ریالی و ۵ ریالی و ۱ ریالی استفاده کند. اما با توجه به مشاهدات بالا با استفاده از این سکه‌ها حداکثر ارزش ۲۴ قابل جمع آوری است. لذا این با $x \geq 25$ متاقض است.

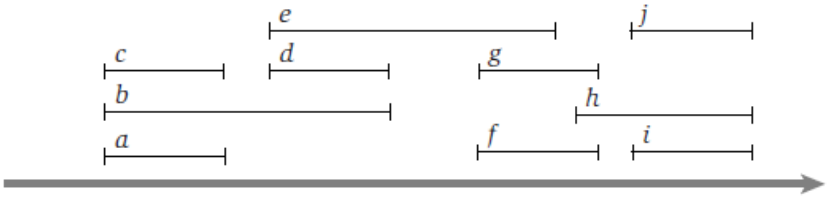
۴. : $100 \leq x$

اگر جواب بهینه از سکه ۱۰۰ ریالی استفاده کرده باشد، با فرض استقرا استراتژی حریصانه برای $x - 100$ درست جواب می‌دهد. پس در کل برای مقدار x استراتژی حریصانه درست است. حال فرض کنید جواب بهینه از سکه ۲۵ ریالی استفاده نکرده باشد. پس تنها راهش این است که از سکه های۲۵ ریالی و ۱۰ ریالی و ۵ ریالی و ۱ ریالی استفاده کند. اما با توجه به مشاهدات بالا با استفاده از این سکه‌ها حداکثر ارزش ۹۹ قابل جمع آوری است. لذا این با $x \geq 100$ متاقض است.

بطور کلی در مسئله صندوقدار سکه‌هایی به ارزش $c_1 < c_2 < \cdots < c_k$ داریم و می‌خواهیم با استفاده از کمترین تعداد سکه مقدار x را خرد کنیم. می‌توان c_i ها را طوری انتخاب کرد که استراتژی حریصانه به جواب بهینه منتهی نشود. برای مثال برای $\{1, 6, 10\}$ و مقدار $x = 12$ استراتژی حریصانه یک سکه ۱۰ ریالی و دو سکه ۱ ریالی بر می‌دارد در حالیکه جواب بهینه دو سکه ۶ ریالی است.

۳ مسئله زمانبندی بازه ها

در مسئله زمانبندی بازه ها، به تعداد n بازه داده شده است. دقت کنید بازه ها ممکن است با هم همپوشانی داشته باشند. هدف انتخاب بیشترین تعداد بازه است بطوریکه بازه های انتخابی همپوشانی نداشته باشند.
در شکل زیر یک نمونه از مسئله داده شده است.

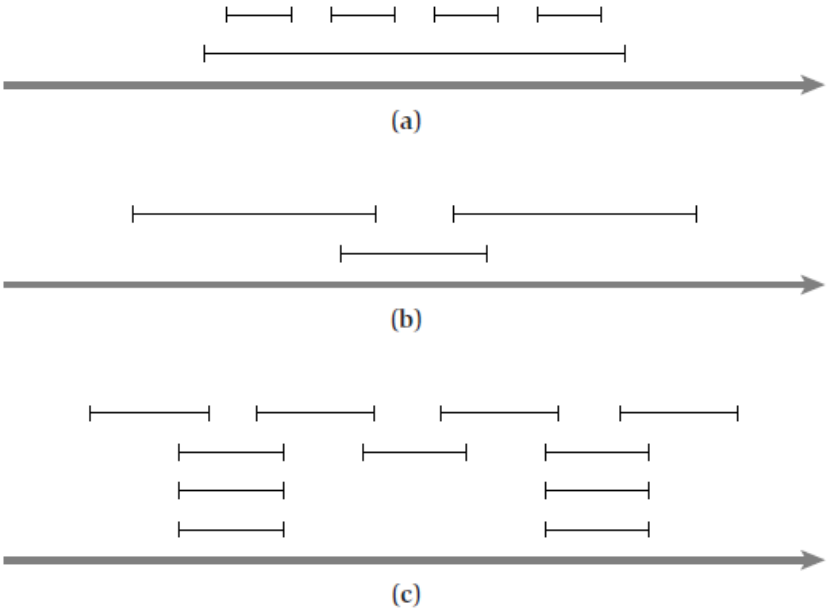


در مثال بالا، یک جواب بهینه میتواند انتخاب بازه های $\{a, d, f, i\}$ باشد. جوابهای بهینه دیگر نیز وجود دارد.

۱.۳ استراتژیهای حریصانه مختلف برای حل مسئله بازه ها

برای حل مسئله بصورت حریصانه غالبا استراتژیهای مختلفی وجود دارد که میتوانیم الگوریتم را بر اساس آنها پی ریزی کنیم. برای مثال، در مسئله بازه ها، استراتژی ما میتواند انتخاب بازه ای باشد که زودتر شروع میشود. یعنی اولویت را به بازه هایی میدهیم که زمان شروع آنها زودتر از بقیه است. اما این استراتژی نمی تواند استراتژی خوبی باشد. به شکل زیر قسمت a توجه کنید. در مثال داده شده، اگر بازه ای که زودتر از همه شروع میشود را برداریم فقط یک بازه برای انتخاب وجود دارد در حالیکه جواب بهینه شامل چهار بازه است.
فرض کنید استراتژی را تغییر دهیم و به بازه هایی اولویت دهیم که کوتاه باشند. اما همانطور که در شکل زیر قسمت b مشاهده میکنید این استراتژی هم نمیتواند جواب بهینه را به ما بدهد.

یک راه دیگر، انتخاب بازه ای است که کمترین تداخل را با بقیه داشته باشد. یعنی بازه ها را از لحاظ تعداد تداخل با بازه های دیگر مرتب کنیم و به ترتیب بازه ای را انتخاب کنیم که تداخل کمتری داشته باشد و همینطور ادامه دهیم. شکل قسمت c نشان میدهد این استراتژی هم جواب بهینه را بدست نمیدهد.

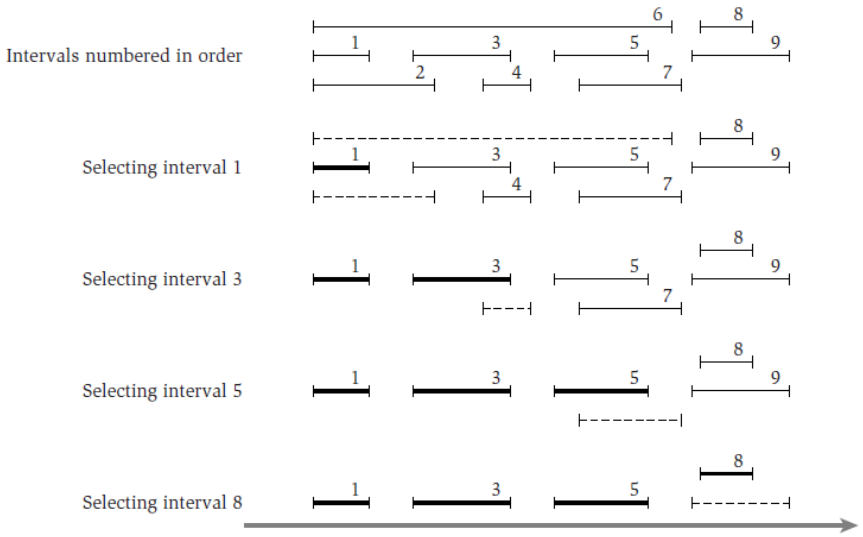


۲.۳ یک استراتژی حریصانه خوب برای مسئله بازه ها

مثالهای نقض بالا نباید ما را در جستجو برای استراتژی حریصانه ناامید کند. هنوز خیلی معیار وجود دارد که امتحان نکرده ایم. یک معیار میتواند انتخاب بازه ها بر اساس زمان خاتمه باشد. یعنی به بازه هایی اولویت بدهیم که زودتر تمام میشوند. بیایید تصور کنیم بازه در واقع یک سری پیشنهاد کار برای یک کارگر هستند. هر کار باید در زمان مشخصی شروع و در زمان مشخصی تمام شود. کارگر اگر بخواهد بیشترین تعداد کار را قبول کند تا سود بیشتری عایدش شود (فرض کنید کارها ارزش یکسان دارند)، طبیعتا کاری را انتخاب میکند که زودتر تمام شود. چون میخواد هر چه زودتر آزاد شود تا کار دیگر را شروع کند. این یک استراتژی معقول است. اتفاقا این استراتژی بهینه است.

میتوانید هر سه مثال بالا را بررسی کنید و ببینید که در هر سه حالت استراتژی انتخاب بازه ای که زودتر تمام میشود جواب بهینه را تولید میکند.

مثال زیر را ببینید. بازه ها بر اساس زمان خاتمه با شماره 1 تا 9 مرتب شده اند. استراتژی حریصانه بر اساس زمان خاتمه، بازه های $1, 3, 5, 8$ را با ترتیب انتخاب میکند.



توصیف الگوریتم حریصانه:

۱. بازه ها را بر اساس زمان خاتمه مرتب کن. فرض $R = (s_1, f_1), \ldots, (s_n, f_n)$ لیست بازه ها بعد از مرتب سازی باشد.

۲. فرض کن A مجموعه بازه های انتخابی باشد. در شروع کار $A = \emptyset$.

۳. از سمت چپ لیست R شروع کن و بازه ای که همپوشانی با بازه های داخل A نداشته باشد را بردار و در مجموعه A قرار بده.

۴. در انتها A جواب الگوریتم است.

توجه کنید که الگوریتم بالا در زمان $O(n \log n)$ قابل پیاده سازی است. زمان اجرا مربوط به قدم اول است که بازه ها در آن مرتب میشوند. بقیه قدم ها در زمان $O(n)$ قابل اجرا هستند.

۳.۳ اثبات بهینه بودن الگوریتم

برای اینکه اثبات کنیم جواب الگوریتم حریصانه بهینه است، یعنی بیشترین تعداد بازه ها انتخاب شده است، از استقرا استفاده میکنیم. فرض کنید O یک جواب بهینه باشد و $o_1, o_2, \ldots, o_m$ لیست بازه های داخل O باشد که بر اساس زمان خاتمه از کوچک به بزرگ مرتب شده اند.

میخواهیم نشان دهیم تعداد بازه های A که جواب الگوریتم حریصانه است برابر با m است. به عبارت دیگر $|A| = m$.

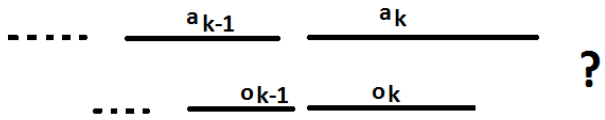
اولا واضح است که بازه های داخل A همپوشانی ندارند. این از طرز کار الگوریتم نتیجه میشود.

حال لم زیر را با استفاده از استقرا اثبات میکنیم.

لم : فرض کنید a_k در لیست بازه های انتخابی توسط الگوریتم k امین بازه باشد. زمان خاتمه a_k کمتر یا مساوی زمانه خاتمه o_k است. به عبارت دیگر

$$\text{finish}(a_k) \leq \text{finish}(o_k)$$

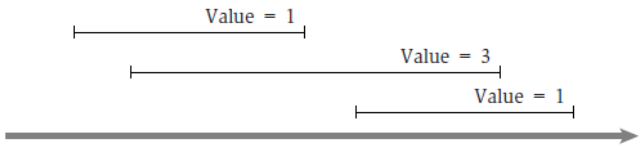
اثبات: اثبات با استقرا. برای پایه استقرا روشن است که $\text{finish}(a_1) \leq \text{finish}(o_1)$
حال برای فرض استقرا، برای هر $i < k$ داریم $\text{finish}(a_i) \leq \text{finish}(o_i)$
با توجه به فرض استقرا، آیا امکان دارد که $\text{finish}(a_k) > \text{finish}(o_k)$. به شکل زیر توجه کنید. این به این معنی است که الگوریتم حریصانه با وجود اینکه امکان انتخاب بازه o_k را داشته اما a_k را انتخاب کرده که زمان خاتمه دورتری دارد. این با منطق الگوریتم حریصانه متناقض است. لذا حتما باید $\text{finish}(a_k)$ کمتر مساوی با $\text{finish}(o_k)$ باشد. □



با داشتن لم بالا، اثبات اینکه $|A| = |O|$ کار ساده ای است. فرض کنید که $|O| > |A|$ و $|A| = r$. این یعنی اینکه بعد از بازه a_r الگوریتم حریصانه نتوانسته بازه ای دیگر را انتخاب کند. اما با توجه به لم بالا می دانیم که $\text{start}(o_{r+1}) > \text{finish}(o_r) \geq \text{finish}(a_r)$ پس الگوریتم حریصانه امکان انتخاب o_{r+1} را داشته و این متناقض با فرض ماست. لذا حتما باید $|A| = |O|$. این نشان میدهد که الگوریتم حریصانه بهینه است.

۴.۳ بازه‌های با ارزش متغیر

همانطور که دیدیم الگوریتم حریصانه، با معیار انتخاب بازه ای که زودتر تمام میشود، جواب بهینه را پیدا میکند. این برای حالتی درست که بازه ها ارزش یکسان داشته باشند. اگر بازه ها از لحاظ ارزش یکسان نباشند و ارزشهای مختلف داشته باشند و بخواهیم یک مجموعه بازه که مجموع ارزش آنها ماکزیمم باشد و همپوشانی هم نداشته باشند را انتخاب کنیم الگوریتم حریصانه مورد بحث جوابگوی مسئله نیست. به مثال زیر توجه کنید. الگوریتم حریصانه اول بازه بالایی را انتخاب میکند و ناچار است که در قدم بعدی بازه پایینی را انتخاب کند که در مجموع ارزش 2 را بدست می دهد. اما جواب بهینه انتخاب بازه وسطی است که ارزش 3 دارد.



در فصول بعدی خواهیم دید که این نسخه از مسئله با استفاده از تکنیک برنامه نویسی پویا قابل حل است.