# آنالیز الگوریتم‌ها (۲۲۸۹۱)

**[بهار ۹۹]**

تمرین سری ۲ — موعد: سه‌شنبه ۶ اسفند ساعت ۱۲

---

- سؤالات خود پیرامون تمرین را با andishe.ghasemi.9@gmail.com مطرح کنید.

### تمرین‌های پیشنهادی (غیرتحویلی)

CLRS 22.1.5, 22.1.6, 22.1.8, 22.2.1, 22.2.7, 33.4.3.

### تمرین‌های تحویلی

۱. روی یک صفحه مختصات $n$ نقطه مشخص شده‌است. به یک وضعیت از $n$ نقطه خوب گوییم اگر به ازای هر جفت نقطه مثل $(x, y)$ و $(x', y')$ یکی از سه شرط زیر برقرار باشد:

۱) $x = x'$

۲) $y = y'$

۳) نقطه‌ی دیگری چون $(x'', y'')$ موجود باشد به طوری‌که $\min(x, x') \leq x'' \leq \max(x, x')$ و $\min(y, y') \leq y'' \leq \max(y, y')$.

می‌خواهیم تعدادی نقطه به مجموعه نقاط اضافه کنیم تا در انتها به یک وضعیت خوب برسیم؛ الگوریتمی طراحی کنید که در زمان $O(n \lg n)$ به مقدار $O(n \lg n)$ نقطه به این نقاط اضافه کند طوری که در انتها یک وضعیت خوب از نقاط داشته‌باشیم.

**راهنمایی برای پاسخ:** اگر شرط اینکه نقطه‌ها متمایز باشند را نداشته‌باشیم، می‌توانیم دقیقا روی هر نقطه، نقطه‌ای قرار داده و در $O(n)$ مسئله را حل کنیم.

اما اگر شرط متمایز بودن نقاط را داشته‌باشیم، الگوریتمی ارائه می‌دهیم که با داشتن مجموعه نقاطی که بر حسب مولفه اول مرتب شده و همان مجموعه نقاط که بر حسب مولفه‌ی دوم مرتب شده، مسئله را حل می‌کند: فرض کنید $n$ نقطه داریم که $(x_1, y_1), \cdots, (x_n, y_n)$ و $x_i \leq x_{i+1}$ و همچنین همین مجموعه نقاط را به صورت $(x'_1, y'_1), \cdots, (x'_n, y'_n)$ و $y'_i \leq y'_{i+1}$ داریم. اگر $n = 1$ باشد، مسئله حل است، فرض کنید $x'$ میانه‌ی $x_i$ها باشد، چون نقاط را مرتب شده داریم $x'$ را با $O(1)$ داریم، پس حداکثر $\frac{n}{2}$ تا از نقاط دارای مولفه‌ی اول کمتر از $x'$ و حداکثر $\frac{n}{2}$ از نقاط دارای مولفه‌ی اول بیشتر از $x'$ هستند، این دو مجموعه را با $O(n)$ محاسبه می‌کنیم و مسئله را به صورت بازگشتی برای قسمت سمت چپ و سمت راست حل می‌کنیم و دو مجموعه نقطه‌ی اضافه‌ی $P_L$ و $P_R$ به‌دست می‌آوریم.

حال آرایه‌ی $A$ را برابر با مولفه‌ی دوم همه‌ی نقاطی که مولفه‌ی اول آن‌ها $x'$ است در زمان $O(n)$ با فور زدن روی $(x'_i, y'_i)$ها می‌سازیم، پس این آرایه یک آرایه‌ی مرتب است. حال $R = \emptyset$ در نظر بگیرید، به ازای هر $i \in \{1, ..., n-1\}$، اگر $y'_i \neq y'_{i+1}$، نقطه‌ی $(x', \frac{y'_i+y'_{i+1}}{2})$ را در صورتی که $\frac{y'_i+y'_{i+1}}{2} \notin A$ به $R$ اضافه می‌کنیم، با توجه به مرتب بودن $A$ و مرتب بودن $\frac{y'_i+y'_{i+1}}{2}$ها، این کار را در مجموع با $O(n)$ می‌توانیم انجام دهیم.

حال $R \cup P_L \cup P_R$ را به عنوان نقاطی که باید اضافه‌کنیم خروجی می‌دهیم.

خب، الگوریتم ارائه شد، ۳ مورد می‌ماند:

(آ) اثبات درستی الگوریتم:
برای اثبات می‌توانیم از استقرا استفاده کنیم، برای $n = ۱$ درستی الگوریتم واضح است، فرض کنید الگوریتم برای همه‌ی $n$ های کمتر از $k$ درست است و برای $n = k$، زوج نقاط به چند دسته تقسیم می‌شوند: یا هردو مولفه‌ی اول کمتر از $x'$ یا هر دو مولفه‌ی اول بیشتر از $x'$ دارند که طبق استقرا نقطه‌ای برای آن اضافه کرده‌ایم. برای مولفه اول یکی کمتر از $x'$ و مولفه اول دیگری بزرگتر یا مساوی $x'$، اگر مولفه‌ی دوم این دو نقطه مساوی باشد، سطر دوم برقرار است و در غیر اینصورت طبق عمل آخر، نقطه‌ای با مولفه‌ی اول $x'$ و مولفه‌ی دوم بین این‌دو اضافه کردیم که شرط سوم برای آن برقرار است و برای حالت چهارم نیز مشابه قبل است و در نتیجه این وضعیت یک وضعیت خوب است و گام استقرا ثابت و حکم استقرا ثابت می‌شود.

(ب) اثبات زمان اجرا: فرض کنید $T(n)$ یک کران‌بالا برای زمان اجرای الگوریتم روی $n$ نقطه باشد که تابعی صعودی است، داریم:

$$T(n) = ۲ \times T(\frac{n}{۲}) + O(n)$$

که می‌دانیم $T(n) = O(n \lg n)$ است.

(ج) اثبات کم بودن تعداد نقاط اضافه‌شده:
مشابه قسمت قبل با استقرا روی $n$ و مشابه قسمت قبل نتیجه می‌شود.

حال ابتدا نقطه‌ها را یک‌بار بر حسب مولفه‌ی اول و یک‌بار بر حسب مولفه‌ی دوم در زمان $O(n \lg n)$ مرتب می‌کنیم و سپس با الگوریتم معرفی شده مسئله را در زمان $O(n \lg n)$ حل می‌کنیم.

۲. برای یک عدد $b$ که $۲ \leq b \leq ۱۰$ است، عدد خوب را عددی طبیعی تعریف می‌کنیم که تمام ارقام آن کمتر از $b$ باشد، مثلا برای $b = ۲$، تمام اعداد با ارقام ۰ و ۱ خوب هستند.
برای $n$ و $b$ داده‌شده، می‌خواهیم با کمک مدل‌سازی اعداد در گراف و استفاده از نسخه‌ای تغییر یافته از «الگوریتم جستجوی اول سطح» کوچکترین عدد خوب مضرب $n$ را پیدا کنیم؛ مثلا برای $n = ۳, b = ۲$ کوچکترین عدد خوب مضرب ۳ برابر با ۱۱۱ است.

(آ) سعی کنید با طراحی یک گراف مناسب، الگوریتمی از $O(n)$ ارائه دهید تا با گرفتن $n, b$ کوچکترین عدد خوب مضرب $n$ را بیابد.
راهنمایی: به ازای باقی‌مانده‌ی هر عدد خوب بر $n$، یک راس در نظر بگیرید.
**راهنمایی برای پاسخ:** یک آرایه $n$ تایی مثل $A$ در نظر می‌گیریم و ابتدا تمام مقادیر آن را ۰ قرار می‌دهیم. یک گراف فرضی در نظر بگیرید که راس‌های آن اعداد خوب است و از راس متناظر عدد $x$ به راس‌های متناظر با اعداد $x۰, x۱, ..., x(b-۱)$ یال جهت‌دار وجود دارد. همچنین برای عدد ۰ نیز یک راس درنظر بگیرید که به راس‌های $۱, ..., b-۱$ یال جهت دار دارد.
الگوریتم $BFS$ را از راس ۰ شروع کنید با این تفاوت که برای علامت‌گذاری به عنوان دیده‌شده از آرایه‌ی $A$ استفاده کنید و برای اضافه‌کردن راس $x$ به صف $BFS$ ، مقدار $x \mod n$ را در آرایه‌ی $A$ یک کنیم و اگر یک بود $x$ را به صف اضافه نکنیم، فقط ۰ را از این قاعده مستثنی کنیم و به اولین راس متناظر عددی مثل $x$ رسیدیم که $x \mod n = ۰$ بود، $x$ را به عنوان جواب خروجی دهیم، همچنین برای راس $x$ همسایه‌های آن را به ترتیب یعنی ابتدا $x۰$، سپس $x۱$ و ... و در آخر $x(b-۱)$ را مورد بررسی برای اضافه‌کردن به صف قرار می‌دهیم.
در واقع، اگر به راسی با باقی‌مانده‌ی تکراری بر $n$ رسیدیم، الگوریتم BFS را از آن راس ادامه نمی‌دهیم.
باید ثابت کنیم که زمان اجرای الگوریتم $O(n)$ است و همچنین خروجی کمترین عدد ممکن است.
با توجه به اینکه هر یال معادل اضافه‌شدن یک رقم است و با توجه به اجرای الگوریتم BFS، پاسخ کوچکترین طول را دارد، همچنین با توجه به اینکه همسایه‌ها را به ترتیب مشاهده کردیم، می‌توان نشان‌داد که به ازای هر باقی‌مانده، کوچکترین‌عدد، اولین مشاهده‌ی ما از راس‌ها در آن باقی‌مانده است.
همچنین زمان اجرای الگوریتم از $O(n)$ است، گرافی که این الگوریتم راس‌ها و یال‌های آن را بازدید می‌کند در نظر بگیرید،

الگوریتم حداکثر $n$ بار راس با باقی‌مانده‌ی جدید می‌بیند و $k$ بار راس باقی‌مانده‌ی تکراری می‌بیند، همچنین راس‌های با باقی‌مانده‌ی تکراری یال ورودی دارند و هیچ یال خروجی ندارند و راس‌های غیرتکراری $b$ یال خروجی دارند، پس در این گراف درجه‌ی هر راس حداقل یک و تعداد یال‌ها برابر با مجموع درجه‌های خروجی و حداکثر $bn$ است، پس تعداد راس‌های گراف از $O(n)$ و تعداد یال‌های گراف نیز از $O(n)$ است و در مجموع الگوریتم BFS به زمان $O(n)$ طول می‌کشد.

(ب) فرض کنید $x$ یک عدد خوب است، همزاد این عدد را عددی مثل $y$ تعریف می‌کنیم که $(x)_b = (y)_{۱۰}$ (منظور از اندیس مبنا است.) مثلا برای $b = ۲$ همزاد عدد خوب ۱۱۱ عدد ۷ است.

الگوریتمی از زمان $O(n^۲)$ ارائه دهید که با ورودی گرفتن $n, b$ کوچکترین عدد خوب که هم خودش و هم همزادش مضرب $n$ باشد بیابید.

راهنمایی: به ازای هر زوج مرتب باقی‌مانده‌ی عدد خوب بر $n$ و همزادش بر $n$، یک راس در نظر بگیرید.

**راهنمایی برای پاسخ:** دقیقا مشابه الف رفتار کنید، ولی اینبار آرایه‌ی $A$ را یک آرایه‌ی دوبعدی $n \times n$ در نظر بگیرید و برای علامت‌گذاری به عنوان دیده شده از باقی‌مانده‌ی $x$ بر $n$ و همزادش بر $n$ استفاده کنید.

بقیه‌ی روند الگوریتم مشابه قسمت قبل است.

موفّق باشید.