

۶ - دنباله‌ای از n عدد a_1, a_2, \dots, a_n به ما داده شده است و از ما خواسته شده است که بزرگ‌ترین زیر دنباله صعودی آن را پیدا کنیم. به یک زیر دنباله صعودی می‌گوییم اگر $\{i_1, i_2, \dots, i_k\}$ زیر مجموعه‌ای از اعداد 1 تا n باشند به گونه‌ای که $i_1 < i_2 < \dots < i_k$ و $a_{i_1} \leq a_{i_2} \leq \dots \leq a_{i_k}$.

الف) بزرگ‌ترین زیر دنباله صعودی برای دنباله زیر را پیدا کنید: (۱ نمره)

5, 3, 4, 8, 7, 10

3, 4, 8, 10

ب) فاطمه پس از کمی فکر کردن، الگوریتم زیر برای حل این مسئله به ذهنش رسید: ابتدا کوچک‌ترین عدد را به عنوان اولین عدد انتخاب کنیم. سپس اولین عدد بزرگ‌تر پس از آن را به عنوان عضو دوم، و سپس اولین عدد بزرگ‌تر از عدد دوم را به عنوان عدد سوم انتخاب کرده و این کار را این‌قدر ادامه دهیم تا دیگر نتوان عدد دیگری را انتخاب کرد. آیا راه‌حل فاطمه همواره بزرگ‌ترین زیر دنباله صعودی را به ما می‌دهد؟ اگر بله، اثبات کنید. اگر نه، یک مثال نقض بزنید. (۲ نمره)

همواره جواب بهینه را به ما نمی‌دهد. به عنوان مثال، در دنباله 1, 2, 3، این الگوریتم تنها عدد ۱ را انتخاب می‌کند، در حالیکه جواب

۹ - رضا به شدت تغییرات قیمت روزانه دلار را دنبال می‌کند و همواره آرزو می‌کند که ای‌کاش می‌دانست قیمت دلار در روزهای آینده چطور تغییر می‌کند تا با خرید و فروش در زمان مناسب، بتواند سودی خوبی به دست آورد. دیشب، وقتی پس از ساعت‌ها مطالعه درس ساختمان داده‌ها برای امتحان امروز خوابش برد، در یک رویای صادقه، جدولی از قیمت دلار در ماه‌های بهمن و اسفند را مشاهده کرد. او ناگهان از خواب پرید و بلافاصله قیمت‌های روزانه را روی یک ورق کاغذ نوشت. حال او فکر می‌کند که چطور می‌تواند با استفاده از ۱۰ میلیون تومان پس‌اندازی که دارد، حداکثر سود ممکن را کسب کند. او می‌خواهد دو روز را انتخاب کند به گونه‌ای که در روز اول (مثلاً ۱۵ بهمن) تمام پس‌انداز خود را به دلار تبدیل کند و در ادامه در روز دوم (مثلاً ۱۳ اسفند) تمام دلار را بفروشد. به او کمک کنید که برای این کار یک الگوریتم طراحی کند. الگوریتم شما باید مقدار قیمت دلار در n روز را بگیرد و در زمان $O(n)$ تعیین کند بهترین روز برای خرید و سپس برای فروش چه روزی است. (۴ نمره)

ابتدا برای هر روز محاسبه می‌کنیم که حداکثر قیمت دلار در روزهای پس از آن، چقدر خواهد بود. فرض کنید که این مقدار را با $m[i]$ نمایش می‌دهیم. اگر قیمت در روز i برابر با $p[i]$ باشد، خواهیم داشت: $m[i] = \max \{ m[i+1], p[i+1] \}$. در یک آرایه دیگر هم نگه می‌داریم که این بیشینه در کدام روز است. حال برای هر روز، حداکثر سود حاصل از خرید در آن روز برابر با $m[i] - p[i]$ است. کافی است این مقدار را برای تمام روزها حساب کرده و روز با مقدار بیشینه را انتخاب می‌کنیم.

۶ - به ما k آرایه مرتب‌شده داده شده است. این k آرایه در مجموع شامل n عدد حقیقی هستند. می‌خواهیم بررسی کنیم که آیا تمام این n عدد متمایز هستند یا نه. الگوریتمی با زمان $O(n \log k)$ و حافظه اضافی $O(k)$ برای این کار ارائه دهید. (۳ نمره)

ابتدا بررسی می‌کنیم که آیا در یکی از آرایه‌ها عدد تکراری وجود دارد یا نه. برای این کار کافی است که هر آرایه را یکبار پیمایش کنیم تا ببینیم آیا دو عنصر متوالی برابر در آن وجود دارد یا نه. این بخش از الگوریتم به زمان $O(n)$ نیاز دارد.

در ادامه بررسی می‌کنیم که آیا عددی در دو آرایه مختلف تکرار شده است یا نه. برای این کار، یک هرم کمینه حداکثر k عنصری ایجاد می‌کنیم که از هر آرایه، یک نماینده دارد. در ابتدا، کوچک‌ترین عنصر هر آرایه را در هرم قرار می‌دهیم. در هر مرحله، ابتدا بررسی می‌کنیم که آیا دو کوچک‌ترین عنصر این هرم با هم برابر هستند یا نه. اگر جواب بله بود، یک عنصر تکراری پیدا کرده‌ایم و کار تمام است. در غیر این صورت، نتیجه این می‌شود که کوچک‌ترین عنصر هرم تنها در یک آرایه آمده است. بنابراین آن را از هرم حذف کرده و عنصر پس از آن در آرایه مربوطه را به هرم اضافه می‌کنیم. اگر هم آرایه تمام شده بود، عنصری را اضافه نمی‌کنیم.

اگر در پایان هرم خالی شد، یعنی هیچ عنصر تکراری در آرایه‌ها وجود ندارد. هر درج و حذف در هرم $O(\log k)$ هزینه دارد و هر یک از n عنصر نیز حداکثر یکبار وارد هرم شده و یکبار از آن خارج می‌شود. پس زمان کلی $O(n \log k)$ خواهد بود.

۴ - دانشجویان درس ساختمان داده ها خیلی اهل فوتبال نیستند. بنابراین، هر کدام از آن ها تنها بخشی از بازی دیروز بین ایران و ازبکستان در جام کافا را به صورت زنده در تلویزیون دیدند. فرض کنید که دانشجوی آام تنها بازه $[s_i, e_i]$ از بازی را به صورت زنده مشاهده کرد. چگونه می توان محاسبه کرد که در بازی دیروز باید حداقل چند گل زده می شد تا تمامی دانشجویان حداقل یک گل را دیده باشند؟ مثلاً اگر ۳ دانشجو در بازه های $[10, 24]$ ، $[14, 20.3]$ و $[32, 44.4]$ بازی را تماشا کرده باشند، باید حداقل ۲ گل (مثلاً در دقایق ۱۵ و ۳۵) زده شده باشد تا همه حداقل یک گل را دیده باشند. زمان اجرای الگوریتم شما چقدر است؟ (۳ نمره)

بازه های تماشای دانشجویان را بر اساس زمان اتمام مرتب می کنیم. دو متغیر هم نگه می داریم: آخرین گل زده شده و تعداد گل ها. سپس از دانشجو یا کوچک ترین زمان اتمام شروع می کنیم. اگر زمان شروع این دانشجو بزرگ تر از آخرین گل زده شده بود، تعداد گل ها را یکی زیاد کرده و آخرین زمان شروع را هم برابر زمان پایان تماشای این بازیکن قرار می دهیم. سپس به سراغ دانشجوی بعدی می رویم. مرتب سازی به زمان $O(n \log n)$ نیاز دارد و پیمایش بعدی نیز در زمان $O(n)$ انجام می شود، پس زمان کلی $O(n \log n)$ است.

اثبات درستی: دانشجو با کم ترین زمان اتمام را در نظر بگیرید. این دانشجو باید یک گل را دیده باشد. پس دیرترین زمانی که این گل می تواند به ثمر رسیده باشد، زمان اتمام تماشای این دانشجو است. اگر گل اول بازی در بهترین جواب، زودتر از این زمان بود، می توان آن را به تاخیر انداخته تا در این زمان زده شود. در این صورت، همچنان این دانشجو و تمام دانشجویان دیگری که آن را قبلاً می دیدند، همچنان می توانند آن را ببینند (و ممکن است تعداد جدیدی نیز ببینند). بنابراین اولین انتخاب ما اشتباه نیست و به همین ترتیب، با استقرا سایر انتخاب های الگوریتم نیز درست خواهد بود.

۴ - یکی از محلات توریستی شهر قزوین، خیابان سفیر امید معروف به رودخانه ی بازار در محله نواب است. نکته دیدنی این محله، خانه های رنگی آن است. شهردار منطقه ۶ که در تعطیلات اخیر به قزوین رفته بود، پس از مشاهده آن، تصمیم می گیرد که جهت افزایش شادی دانشجویان دانشگاه امیرکبیر، n ساختمان شمال خیابان رشت را نیز با استفاده از سه رنگ قرمز، سبز، و آبی رنگ کند. پس از یک بررسی اولیه و با توجه به قیمت هر یک از این رنگ ها و نمای خانه ها، هزینه لازم برای رنگ کردن هر خانه به هر کدام از این رنگ ها آماده شده است (هزینه رنگ کردن خانه آام به رنگ قرمز، آبی، و سبز به ترتیب برابر بی $CR[i]$ ، $CB[i]$ و $CG[i]$ است). روشی ارائه دهید که کم ترین هزینه لازم برای رنگ کردن ساختمان ها را به گونه ای که هیچ دو ساختمان مجاور هم رنگ نباشند. زمان الگوریتم خود را هم بیان کنید. (۴ نمره)

سه مقدار $R[i]$ ، $G[i]$ و $B[i]$ را به این صورت تعریف می کنیم که هر یک کمترین هزینه رنگ کردن آ خانه اول باشد به گونه ای که خانه آام قرمز، سبز، و یا آبی باشد. آنگاه رابطه محاسبه بازگشتی این سه تابع به صورت زیر خواهد بود:

$$R[i+1] = CR[i] + \min \{G[i], B[i]\} \quad B[i+1] = CB[i] + \min \{G[i], R[i]\} \quad G[i+1] = CG[i] + \min \{R[i], B[i]\}$$

همچنین خواهیم داشت که $R[0]=B[0]=G[0]$. به این ترتیب، هر یک از این مقادیر در زمان $O(1)$ از روی مقادیر قبلی محاسبه می شوند. پس در کل، در زمان $O(n)$ هر سه تابع برای تمام مقادیر آ از ۱ تا n محاسبه می شود. جواب مسئله نیز $\max\{R[n], G[n], B[n]\}$ است زیرا خانه آخر یکی از سه رنگ را خواهد داشت.

۳- الگوریتمی ارائه دهید که در بازی مار و پله کمترین تعداد دفعات ریختن تاس برای برنده شدن بازی را محاسبه کند.

۳- یک گراف جهت دار که هر رأس آن یک خانه در بازی مار و پله را نشان می دهد با شرایط زیر رسم می کنیم:

- از هر رأس v به هر یک از رأس های $v+1, v+2, \dots, v+6$ یک یال رسم می کنیم ؛ در صورتی که در آن رأس پله یا مار نباشد.
- اگر در هر یک از رأس های $v+1, v+2, \dots, v+6$ پله یا مار به رأس x وجود داشت ، به جای رسم یال از v به آن رأس ، از v به x یال رسم می کنیم.

بعد از انجام این مراحل روی تمام رئوس گراف، با استفاده از BFS کوتاه ترین مسیر از رأس ابتدایی به رأس انتهایی را پیدا می کنیم.

۱- کشوری با n شهر و m جاده دو طرفه داریم هر جاده دو شهر را به هم متصل می کند در این کشور دزدی زیاد شده است و هدفمان دستگیری دزدها است برای همین می توانیم در یک شهر پلیس قرار دهیم و یک شهر امن است اگر و تنها اگر پلیسی وجود داشته باشد که آن پلیس بتواند با عبور از جاده ها به شهر مدنظر برسد. از آنجایی که قرار دادن پلیس در هر شهر هزینه دارد می خواهیم با کمترین هزینه ممکن همه شهر های کشور را امن کنیم حداقل به چند پلیس نیاز داریم برای یک کشور که به ما داده شده است. سوال باید در $O(n+m)$ حل شود.

۱- کافی است کشور را به صورت یک گراف بدون جهت در نظر بگیریم و تعداد مولفه های گراف را بدست آوریم. مولفه: مجموعه ای از راس ها در یک مولفه اند اگر و تنها اگر از هر کدام به دیگری مسیری وجود داشته باشد.

یک آرایه به اندازه n از ۰ و ۱ در نظر می گیریم و از شهر ۱ شروع کرده تا شهر n پیش می رویم هر کدام از راس هایی را که رویشان حلقه انداختیم اگر از قبل مقدار آرایه برای آن شهر که گرفتیم ۱ باشد کاری نمی کنیم در غیر این صورت از آن شهر dfs می زنیم و وارد هر راسی که شدیم مقدار آرایه مربوط به آن راس را ۱ می کنیم و در انتها یک واحد به جواب می افزاییم این الگوریتم تعداد مولفه ها را بدست می آورد که همان جواب سوال است.

۳ - می خواهیم در دانشکده، یکروز را به صورت گروهی به عنوان جشن تولد برگزار کنیم. برای این منظور، از دانشجویان دانشکده خواستیم که روز تولد خود را برای ما ارسال کنند. حال می خواهیم روزی را که بیشترین تعداد دانشجو در آن متولد شده اند را پیدا کنیم. اگر دانشکده n دانشجو داشته باشد، چگونه می توانیم در زمان $O(n)$ این کار را انجام دهیم؟ (۱ نمره)

می دانیم که هر دانشجو در یکی از ۳۶۵ روز سال متولد شده است. پس می توانیم روزهای تولد دانشجویان را با مرتب سازی شمارشی در زمان $O(n)$ مرتب کنیم (چون تعداد روزها محدود است). در حین این کار، می توانیم بین تعداد تکرارهای هر روز، عدد بیشینه را هم پیدا کنیم که نیاز به یک کار $O(n)$ اضافه دارد.

۱ - حسن برنامه ای برای محاسبه ارزان ترین روش مسافرت به شهرهای ایران با استفاده از اتوبوس طراحی کرده است. خروجی این برنامه در قالب یک جدول شامل کمترین هزینه برای رفتن به هر شهر از تهران است. امروز صبح به حسن خبر داده اند که قیمت بلیط اتوبوس در تمامی مسیرها ۷۵٪ افزایش یافته است. اگر تعداد شهرها برابر با n و تعداد مسیرهای اتوبوسی بین شهرها m باشد، به روزرسانی این جدول به چه زمانی نیاز خواهد داشت؟ (۱ نمره)

اگر قیمت بلیط تمام مسیرها به یک نسبت افزایش یافته است، کوتاه ترین مسیر قبلی همچنان نیز کوتاه ترین مسیر است و تنها قیمت آن به همان نسبت افزایش یافته است. بنابراین با افزایش ۷۵٪ تمام مسیرها، هزینه مسافرت به هر شهر نیز ۷۵٪ افزایش می یابد و کافی است که تمامی اعداد این جدول در ۱.۷۵ ضرب شود. این کار در زمان $O(n)$ قابل انجام است.