



## طراحی الگوریتم - پاسخ نامه تکلیف دوم

- الگوریتم‌های تقسیم و غلبه متفاوتی که خروجی صحیح داشته باشند، نمره سوال را دریافت می‌کنند.
- در صورت وجود هرگونه ابهام می‌توانید در گروه تلگرام یا گروه اسکایپ سوالات خود را مطرح کنید.
- از طریق ایمیل زیر می‌توانید با TA مربوط به این تکلیف در ارتباط باشید.

mnaeimi+algo@ec.iut.ac.ir -

- تصحیح و نمره‌دهی این تکلیف به زودی انجام می‌شود.

سوال ۱. (۱۵ نمره) فرض کنید در بازار بورس میزان سود هر فرد براساس مجموع درصد سودهای سهم از روز ورود به سهم تا روز خروج از سهم به دست می آید. آرایه‌ای شامل سود روزهای مختلف که سودها، مثبت و ضررها، منفی هستند، به شما داده می‌شود. الگوریتم تقسیم و غلبه‌ای به دست آورید که بیشترین میزان مجموع سودی که فرد می‌توانست به دست آورده باشد را حساب کند.

• ایده اصلی: Maximum Subarray Sum using Divide and Conquer algorithm

• پاسخ: می‌خواهیم بزرگترین مجموع زیرآرایه پیوسته را محاسبه کنیم.

– ابتدا آرایه داده شده را به دو نیمه تقسیم می‌کنیم.

– بزرگترین مقدار یکی از سه مقدار زیر، نتیجه بزرگترین مجموع زیرآرایه پیوسته است:

\* بزرگترین مجموع زیرآرایه پیوسته نیمه سمت چپ (به صورت بازگشتی).

\* بزرگترین مجموع زیرآرایه پیوسته سمت راست (به صورت بازگشتی).

\* بزرگترین مجموع زیرآرایه پیوسته که شامل نقطه میانی باشد (شروع در نیمه چپ و پایان در نیمه راست).

دو مقدار اول به صورت بازگشتی به سادگی محاسبه می‌شوند. اما برای محاسبه مقدار سوم در زمان خطی ابتدا بزرگترین مجموع زیرآرایه پیوسته از نقطه میانی به سمت چپ (در نیمه سمت چپ) را محاسبه می‌کنیم، سپس بزرگترین مجموع زیرآرایه پیوسته از نقطه میانی (به صورت دقیق‌تر از  $Mid + 1$ ) به سمت راست (در نیمه سمت راست) را محاسبه می‌کنیم و در نهایت این دو مقدار را با یکدیگر جمع می‌کنیم و برمی‌گردانیم.

سوال ۲. (۱۵ نمره) می‌دانیم که مسیر یک جاده بین دو شهر که دو طرف کوه قرار دارند به صورتی است که از ابتدای جاده تا نقطه‌ای از مسیر، سربالایی و از آن نقطه تا انتهای جاده به صورت سربایینی می‌باشد. ارتفاع نقاط مختلف این جاده به ترتیب در آرایه‌ای از اعداد به شما داده می‌شود. الگوریتم تقسیم و غلبه‌ای به دست آورید تا در زمان  $O(\log n)$  محل مرتفع‌ترین نقطه این جاده را به دست آورده.

• ایده اصلی: Binary Search in Unimodal Array

• پاسخ: برای حل این سوال از جستجوی دودویی استفاده می‌کنیم.

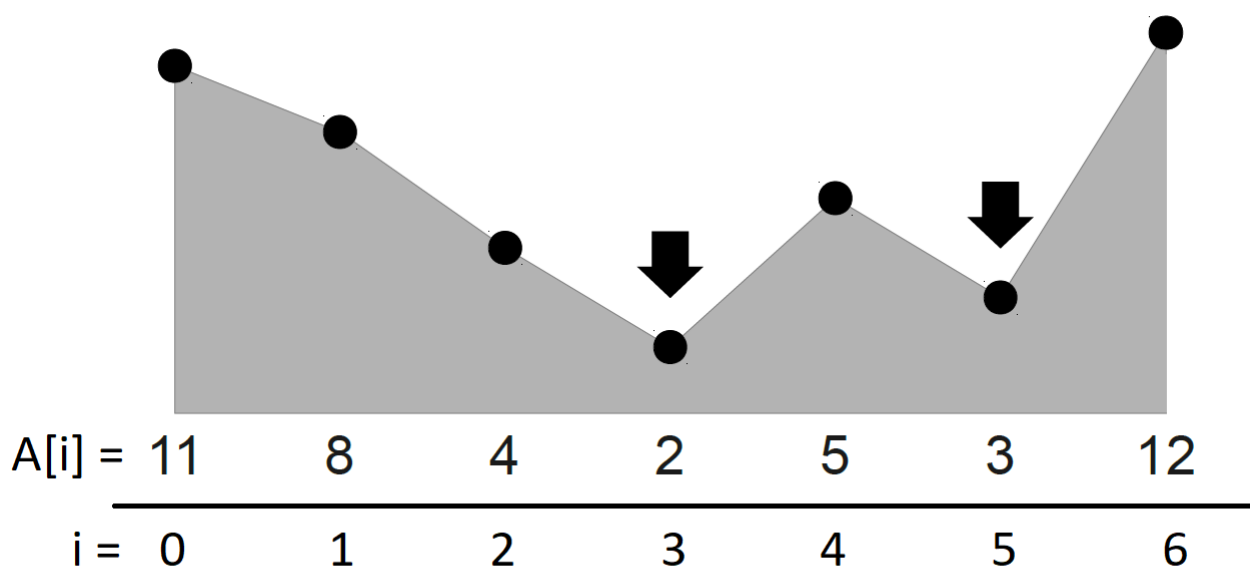
– در آرایه داده شده نقطه میانی ( $A[n/2]$ ) را مورد بررسی قرار می‌دهیم:

\* در صورتی که نقطه مورد بررسی از دو همسایه سمت چپ و راست خود بزرگتر بود، آن را به عنوان قله برمی‌گردانیم.

\* در صورتی که نقطه مورد بررسی از همسایه سمت چپ خود کوچکتر و از همسایه سمت راست خود بزرگتر بود، نیمه سمت چپ آرایه را به صورت بازگشتی مورد بررسی قرار می‌دهیم.

\* در صورتی که نقطه مورد بررسی از همسایه سمت چپ خود بزرگتر و از همسایه سمت راست خود کوچکتر بود، نیمه سمت راست آرایه را به صورت بازگشتی مورد بررسی قرار می‌دهیم.

سوال ۳. (۲۰ نمره) می‌خواهیم محلی برای جمع آوری آب باران در یک منطقه بسازیم. ارتفاع نقاط مختلف زمین به صورت آرایه ای یک بعدی در اختیار شما قرار می‌گیرد. هیچ دو ارتفاع مجاور ی‌کسان نیستند. نقطه‌ای برای این کار مناسب است که نسبت به دو نقطه مجاورش ارتفاع کمتری داشته باشد. الگوریتم تقسیم و غلبه ای بنویسید که یک نقطه مناسب را بیابد.



به طور مثال در آرایه فوق نقاط ۳ و ۵ محل جمع شدن آب می باشند.

• ایده اصلی: Find a Peak Element

• پاسخ: برای حل این سوال از جستجوی دودویی استفاده می کنیم.

- ابتدا بر اساس تکنیک جستجوی دودویی بررسی می‌کنیم که نقطه میانی، نقطه مناسب است یا خیر.

- در صورتی که نقطه میانی نقطه مناسب نبود:

\* در صورتی که همسایه سمت چپ کوچکتر از نقطه میانی بود، نیمه سمت چپ آرایه را به صورت بازگشتی مورد بررسی قرار می دهیم.

\* در غیر این صورت نیمه سمت راست آرایه را به صورت بازگشتی مورد بررسی قرار می‌دهیم.

توجه کنید که در این سوال تضمین می‌شود چنین نقطه‌ای وجود دارد و ترتیب بررسی همسایه سمت چپ یا راست می‌تواند متفاوت باشد. این بررسی انجام می‌شود تا در صورتی که نقطه میانی کوچکتر از یکی از همسایه‌ها بود و بزرگتر از همسایه دیگر، ادامه حل مساله از سمت همسایه کوچکتر پیش برود، زیرا همسایه بزرگتر ممکن است در نیمه خود (زیر آرایه‌ای که در آن قرار دارد) کوچکتر از همسایه‌ی دیگر خود باشد و به عنوان یک جواب اشتباه در خروجی برگردانده شود.

سوال ۴. (۲۵ نمره) تعداد زیادی سکه که تعداد سکه‌ها توانی از عدد ۳ است و یک ترازوی دوکفه‌ای در اختیار داریم. شما می‌توانید هر تعداد سکه را در هر کفه قرار دهید. در میان سکه‌ها دقیقا یک سکه وزن متفاوتی (ممکن است بیشتر یا کمتر باشد!) با بقیه سکه‌ها دارد و وزن بقیه سکه‌ها کاملا یکسان است. الگوریتم تقسیم و غلبه‌ای بنویسید که با کمترین تعداد مقایسه سکه متفاوت را بیابد. پیچیدگی زمانی آن را بررسی نمایید.

• ایده اصلی: Divide Coins into Three Piles

• پاسخ: ابتدا سکه‌ها را به سه مجموعه جدا تقسیم می‌کنیم و دو مجموعه را توسط ترازو مورد بررسی قرار می‌دهیم:

– در صورتی که دو مجموعه وزن برابری دارند، سکه متفاوت در مجموعه سوم قرار دارد.

\* در اولین مقایسه یکی از دو مجموعه اولیه را با مجموعه سوم مقایسه می‌کنیم تا بفهمیم سکه متفاوت وزن بیشتری دارد یا کمتر.

– در صورتی که یکی از دو مجموعه سبک‌تر بود، تنها در اولین مرحله آن را با مجموعه سوم نیز مقایسه می‌کنیم:

\* صورتی که از مجموعه سوم نیز سبک‌تر بود، سکه متفاوت سبک‌تر می‌باشد و در مجموعه سبک‌تر قرار دارد.

\* در غیر این صورت، سکه متفاوت سنگین‌تر می‌باشد و در مجموعه سنگین‌تر قرار دارد.

– مرحله به مرحله مقایسه‌ها را انجام می‌دهیم تا سکه متفاوت را بیابیم.

پیچیدگی زمانی الگوریتم فوق با توجه به اینکه در هر مرحله یک سوم سکه‌ها را مورد بررسی قرار می‌دهیم و تنها یک مقایسه برای فهمیدن کمتر یا بیشتر بودن وزن سکه متفاوت انجام می‌شود،  $O(\log n + 1) = O(\log n)$  می‌باشد.

سوال ۵. (۲۵ نمره) مجموعه نقاط  $S$  که شامل نقاطی به فرم  $(x, y)$  که  $x$  و  $y$  اعداد حقیقی هستند، در یک صفحه دوبعدی به شما داده شده است. یک نقطه مسلط  $(x_1, y_1)$ ، نقطه‌ای است که هیچ نقطه دیگری مانند  $(x_2, y_2)$  در صفحه وجود نداشته باشد که  $x_2 \geq x_1$  و  $y_2 \geq y_1$  (هر دو شرط به صورت همزمان) برقرار باشند. الگوریتم تقسیم و غلبه‌ای به دست‌آورد که مجموعه نقاط مسلط مجموعه  $S$  را در زمان  $O(n \log n)$  پیدا کند.

• ایده اصلی: The Maxima-Set

• پاسخ: این مسئله ناشی از بهینه‌سازی چند جانبه است، جایی که ما علاقه مند به بهینه‌سازی گزینه‌هایی هستیم که به متغیرهای متعددی بستگی دارند. به عنوان مثال، خرید یک تلفن همراه با کمترین قیمت و بهترین مشخصات ممکن. نقطه‌هایی که عضو مجموعه ماکسیما نیستند، از آنجا که نقاط دیگری در  $S$  بر آن‌ها تسلط دارند، قابل حذف هستند. بنابراین یافتن مجموعه نقاط ماکسیما می‌تواند به عنوان نوعی فیلتر عمل کند که برای انتخاب‌های بهینه، فقط آن نقاطی را که باید کاندید شوند انتخاب می‌کند.

-  $S$  را به ترتیب مناسب مرتب می‌کنیم. ترتیب مناسب می‌تواند با توجه به مسئله باشد، مثلاً ابتدا به ترتیب  $x$  ها و سپس به ترتیب  $y$  ها.

-  $S$  را با استفاده از یک خط عمودی در نقطه میانی تقسیم می‌کنیم.

- به صورت بازگشتی مسئله ماکسیما را برای مجموعه نقاط سمت چپ و همچنین نقاط سمت راست، حل می‌کنیم.

- در گام ادغام:

\* نقاط ماکسیما مجموعه سمت راست، نقاط ماکسیما برای مجموعه  $S$  نیز می‌باشند.

\* اما برخی از نقاط ماکسیمای مجموعه چپ ممکن است تحت تسلط سمت چپ‌ترین نقطه ماکسیما از مجموعه ماکسیمای سمت راست قرار بگیرند.

\* بنابراین مجموعه ماکسیمای سمت چپ را اسکن می‌کنیم و نقاطی را که تحت تسلط این نقطه هستند را حذف می‌کنیم.

\* اجتماع مجموعه ماکسیمای باقی مانده از سمت چپ و مجموعه ماکسیمای سمت راست، مجموعه ماکسیمای  $S$  را تشکیل می‌دهند.

