فصل یکم

پنج‌شنبه‌ی سیزدهم

اثبات کنید در جهان بهترین هستید...

کوشش مثال زدنی دو هفته‌ی اخیر برای حل مسئله‌ی ارائه شده در پنج‌شنبه‌های یازدهم و دوازدهم ما را (که برای استقبال گسترده از فصل جدید پنج‌شنبه‌های سخت کاملا آماده بودیم) شگفت زده کرده است (چه کسی را شگفت زده نکرده است؟). با غرور و افتخار شاهد بودیم که همه‌ی اعضای تیم فنی (و احتمالا بیشتر خوانندگانی که پنج‌شنبه‌های سخت را دنبال می‌کنند) برای ارائه شدن بهترین الگوریتم جهان برای این مسئله لحظه شماری می‌کنند. فقط در هفته‌ی گذشته، دال سه بار آخرین نتیجه‌ها و بهترین جواب‌های ارائه شده برای این مسئله را در جلسه‌های تیم فنی شرح داده است. اشتیاق اعضای تیم فنی تا حدی است که هر یک به هر شکل ممکن بارها در شبانه روز با دال تماس می‌گیرند تا از آخرین اخبار در مورد پنج‌شنبه‌ی یازدهم و دوازدهم آگاه شوند. تعداد نامه‌ها، تماس‌های تلفنی و ملاقات‌های دال در هفته‌ی اخیر به شدت افزایش یافته است و فشار زیاد پاسخ دادن به این درخواست‌ها، سلامتی دال را تهدید می‌کند (ما نیز، در کنار خوانندگان محترم، در این مورد بسیار نگران هستیم). در این شرایط، در آخرین جلسه‌ی تیم فنی، با پیشنهاد دال تصمیم گرفته شد مسئله‌ی جدیدی در این پنج‌شنبه منتشر نشود تا وقت بیشتری به مسئله‌ی هفته‌ی گذشته اختصاص یابد. از این رو، ما نیز به جای انتشار مسئله‌ی جدید، یادداشت‌های گرانبهای دال در مورد مسئله‌ی پنج‌شنبه‌ی یازدهم را بدون هیچ تغییری برای خوانندگان گرامی تکرار می‌کنیم.

یادداشت‌های دال در مورد نقطه‌های هم‌خط

برای یافتن نقطه‌های هم‌خط، از الگوریتم‌های زیر می‌توان استفاده کرد:

الگوریتم اول: به ازای هر دو نقطه از نقطه‌های ورودی، تعداد نقطه‌هایی محاسبه شود که از روی خط عبور کننده از آن دو نقطه می‌گذرند. در محاسبه‌ی خط‌ها باید به این نکته توجه کرد که هیچ خطی نباید دو بار گزارش شود. پیچیدگی زمانی این الگوریتم $O(n^۳)$ است و نه تنها برای پنج‌شنبه‌ی دوازدهم، برای پنج‌شنبه‌ی یازدهم نیز کافی نیست.

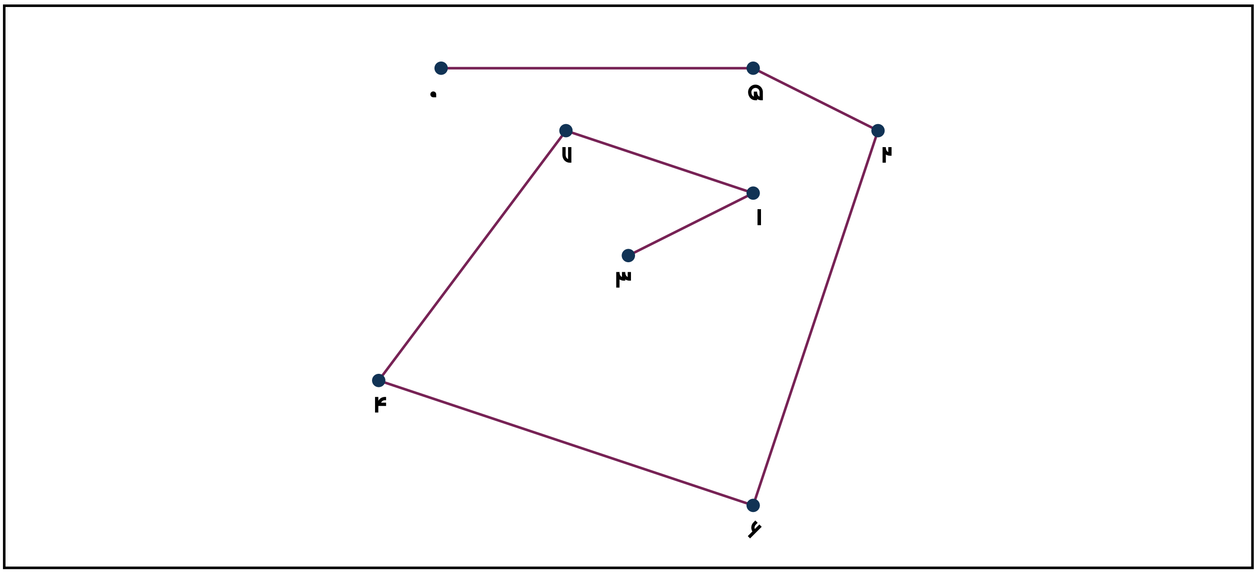
الگوریتم دوم: به ازای دو نقطه از نقطه‌های ورودی، شیب و عرض از مبدأ خطی که از آن دو می‌گذرد، محاسبه گردند و در یک جدول درهم‌سازی وارد شوند. همه‌ی نقطه‌های هم‌خط، خط‌هایی را تشکیل می‌دهند که شیب و عرض از مبدأ یکسانی دارند. به دلیل $O(n^۲)$ عمل افزودن و پرسش از جدول درهم‌سازی، این الگوریتم در عمل شاید چندان مناسب نباشد.

الگوریتم سوم: در این الگوریتم هر نقطه به صورت مجزا در نظر گرفته می‌شود و سایر نقطه‌ها با توجه به زاویه‌ی آنها دور آن نقطه مرتب می‌شوند. در این ترتیب، نقطه‌های هم‌خط، پشت سر هم قرار می‌گیرند. پیچیدگی زمانی این الگوریتم $O(n^۲ \log n)$ است. با بهبود الگوریتم مرتب‌سازی نقطه‌ها بر اساس زاویه، پیچیدگی این الگوریتم می‌تواند به $O(n^۲)$ تبدیل شود.

الگوریتم چهارم: نقطه‌ها به خط‌هایی در صفحه‌ی دوگان (Dual Plane) نگاشت می‌شوند. سه نقطه‌ی هم‌خط در صفحه‌ی دوگان از یک نقطه‌ی مشترک می‌گذرند. بنابراین کافی است نقطه‌های برخورد خط‌ها در صفحه‌ی دوگان بررسی شوند.

بهبود مرتب‌سازی

به نظر می‌رسد بهبود الگوریتم مرتب‌سازی برای تشخیص نقطه‌های هم‌خط، مسئله‌ی آسانی نباشد. با این وجود، روزنه‌های امیدی برای آن وجود دارند. درباره‌ی چینش نقاط به شکلی که تغییر ترتیب نقطه‌ها تا حد امکان حداقل باشد، یکی از ایده‌هایی که به ذهنم رسید انتخاب چپ‌ترین نقطه‌ی انتخاب نشده از نقطه‌ی قبلی است. برای نمونه، شکل زیر را در نظر بگیرید: نقطه‌های انتخاب شده در الگوریتم از صفر شروع می‌شوند و تا سه (به ترتیب خط کشیده شده) پیش می‌روند (این مسئله شباهت زیادی به مسئله‌ی «Convex Hull» دارد؛ کمی مشابه یافتن تعدادی چندضلعی محیطی تو در تو).



اگر به صورت نادقیق بیان شود، هر دور از این حلقه با حداکثر $O(n^۲)$ خط از نقطه‌ها (که از اتصال هر دو نقطه ایجاد می‌شود) تلاقی می‌کند. هر یک از این تلاقی‌ها ترتیب دو نقطه در مرتب‌سازی بر اساس زاویه را عوض می‌کنند. بنابراین برای مرتب‌سازی نقاط برای همه‌ی نقطه‌های پایه، $O(h \cdot n^۲)$ جابجایی انجام می‌شود که در آن h تعداد حلقه‌های تو در توی مسیر است. اما نکته‌ای که در پیاده‌سازی این ایده اهمیت دارد این است که برای مرتب‌سازی بر حسب زاویه، نباید زاویه‌های بیش از ۱۸۰ درجه به قبل از آن انتقال یابند (در غیر این صورت، ترتیب نقطه‌های مرتب شده بر اساس زاویه تغییرات زیادی خواهند داشت). این روش کمی الگوریتم تشخیص نقطه‌های هم‌خط را پیچیده می‌کند ولی در شرایطی که h کوچک باشد، الگوریتم بسیار سریع‌تر خواهد شد.