حل سوال ۲۴: وضعیت LIS

صورت سوال

یک آرایه به طول n داریم که مقدار عضو iـام آن برابر با a_i است. شما باید الگوریتمی با پیچیدگی زمانی $\mathcal{O}(n \log n)$ ارائه دهید که به ازای هر $1 \leq i \leq n$ مشخص کند که اندیس iـام آرایه در کدام گروه زیر قرار می‌گیرد:

- **گروه ۱:** اندیس‌هایی که در تمام LIS های آرایه ظاهر شده‌اند.
- **گروه ۲:** اندیس‌هایی که در بعضی ولی نه همه LIS های دنباله ظاهر شده‌اند.
- **گروه ۳:** اندیس‌هایی که در هیچکدام از LIS های دنباله ظاهر نشده‌اند.

محاسبه LIS پیشوندی و پسوندی

گام ۱ و ۲: یافتن طول زیردنباله‌ها

آرایه $f[i]$ را به عنوان طول LIS ختم شده به a_i و آرایه $b[i]$ را به عنوان طول LIS شروع شده از a_i تعریف می‌کنیم.

این مقادیر را می‌توان با استفاده از برنامه‌نویسی پویا (DP) در زمان $\mathcal{O}(n \log n)$ محاسبه کرد. برای آرایه پسوندی، کافیست آرایه را معکوس و عناصر را قرینه کنیم و همان الگوریتم را اجرا نماییم.

طول کل و شرط حضور در LIS

گام ۳: یافتن طول کل LIS

طول ماکزیمم دنباله برای کل آرایه (L) برابر است با:

$$L = \max_{۱ \le i \le n} f[i]$$

شرط حضور یک عنصر در LIS:

هر زیردنباله صعودی که شامل a_i باشد، از ترکیب LIS پیشوندی و پسوندی متصل به a_i ساخته می‌شود. چون خود a_i در هر دو شمرده شده، بیشترین طول ممکن برای دنباله‌ای که از آن می‌گذرد برابر است با $f[i] + b[i] - ۱$.

اثبات درستی دسته‌بندی - گروه ۳

چرا عناصر با $f[i] + b[i] - ۱ < L$ در گروه ۳ هستند؟

اگر برای عنصری داشته باشیم $f[i] + b[i] - ۱ < L$، به این معناست که حتی اگر طولانی‌ترین دنباله صعودی ختم شده به a_i را به طولانی‌ترین دنباله صعودی شروع شده از a_i متصل کنیم، طول کل دنباله حاصل از L کمتر خواهد بود.

در نتیجه، هیچ دنباله صعودی به طول L وجود ندارد که از عنصر a_i عبور کند، بنابراین این عنصر در هیچ LISای قرار ندارد.

آرایه شمارش (Frequency Array)

گام ۴: شمارش جایگاه‌ها

برای تمایز عناصر گروه ۱ و ۲، باید بدانیم آیا مسیرهای جایگزینی برای جایگاه $f[i]$ در یک LIS وجود دارد یا خیر.

یک آرایه شمارنده (count) تعریف می‌کنیم. برای هر عنصری که پتانسیل حضور در LIS را دارد (یعنی $f[i] + b[i] - ۱ = L$)، یک واحد به $\texttt{count}[f[i]]$ اضافه می‌کنیم. این آرایه نشان می‌دهد برای جایگاه kـام در LIS، چند کاندیدای معتبر در کل آرایه وجود دارد.

اثبات درستی دسته‌بندی - گروه‌های ۱ و ۲

تشخیص گروه ۱ (حضور در تمام LISها):
اگر $\texttt{count}[f[i]] = ۱$ باشد، یعنی a_i تنها عنصر در کل آرایه است که می‌تواند جایگاه $f[i]$ را در یک LIS معتبر پر کند. چون هر LIS به طول L باید دقیقاً یک عنصر در جایگاه $f[i]$ داشته باشد، تمام LISها **مجبورند** از a_i عبور کنند.

تشخیص گروه ۲ (حضور در بعضی LISها):
اگر $\texttt{count}[f[i]] > ۱$ باشد، یعنی عناصر دیگری نیز وجود دارند که می‌توانند همین جایگاه را پر کنند. پس مسیرهای جایگزینی وجود دارند و a_i تنها در **بعضی** از LISها حضور دارد.

تحلیل پیچیدگی

پیچیدگی زمانی:

- محاسبه آرایه‌های f و b با استفاده از DP و جستجوی دودویی هر کدام $\mathcal{O}(n \log n)$ زمان می‌برد.
- یافتن طول L، پر کردن آرایه `count` و تعیین گروه هر عنصر، هر کدام با یک پیمایش خطی در زمان $\mathcal{O}(n)$ انجام می‌شوند.
- **زمان کل:** $\mathcal{O}(n \log n)$

پیچیدگی حافظه:

- آرایه‌های f، b و آرایه شمارنده، همگی طول n دارند و به حافظه خطی نیاز دارند.
- **حافظه کل:** $\mathcal{O}(n)$