تورنمنت شطرنج

نگار یاراحمدی

صورت سوال

در مسابقات شطرنج هر ۲ بازیکن با یکدیگر مسابقه می‌دهند و هر بازیکن در صورت برد هر بازی ۲ امتیاز، در صورت تساوی ۱ امتیاز و در صورت باخت صفر امتیاز می‌گیرد. گری کاسپاروف، در دفترچه‌ی خاطراتش همواره جدول نهایی امتیازات مسابقات را به‌ترتیبِ امتیاز نوشته است، ولی علاقه‌ای به نوشتن ریز نتایج بازی‌ها نداشته است.

هدف ما این است که ببینیم این نوشته‌ها حقیقی هستند یا خیر. برای این کار می‌خواهیم ببینیم که آیا جدولی از نتایج مسابقات وجود دارد که جدول امتیازات نوشته‌شده متعلق به آن باشد؟

فرض‌کنید تعداد بازیکن‌ها برابر n است، برای پیداکردن چنین جدولی الگوریتمی از زمان چندجمله‌ای ارائه‌دهید.

فهم سوال

n نفر شطرنج‌باز داریم. همه با هم بازی می‌کنند.
قانون امتیازدهی به این صورت است:

- **برد:** 2 امتیاز برای برنده، 0 امتیاز برای بازنده. (جمع امتیازات توزیع شده = 2)
- **مساوی:** 1 امتیاز برای هر دو نفر. (جمع امتیازات توزیع شده = 2)

توجه کنید که در هر یک مسابقه، تحت هر شرایطی دقیقا 2 امتیاز بین دو بازیکن پخش می‌شود.
حال فقط یک لیست از امتیازات نهایی به ما داده شده است، ما می‌خواهیم بفهمیم آیا می‌شود جدولی از برد و باخت و مساوی ساخت که در نهایت بازیکنان دقیقا همین امتیازات را بگیرند؟

راه حل: مدل‌سازی با شار شبکه (Network Flow)

ما این مسئله را به یک شبکه از لوله‌های آب تشبیه می‌کنیم. می‌خواهیم ببینیم آیا می‌توانیم مقدار مشخصی "امتیاز" را از یک مبدا به یک مقصد طوری هدایت کنیم که شرایط مسئله نقض نشود.
برای این کار یک گراف (شبکه) با اجزای زیر می‌سازیم:

گره‌ها (Nodes):

- **گره مبدا (S):** منبع اصلی تولید امتیازات.
- **گره مقصد (T):** جایی که در نهایت امتیازات باید آنجا جمع شوند.
- **گره‌های مسابقات:** به ازای هر مسابقه بین دو بازیکن i و j، یک گره به اسم M_{ij} می‌سازیم. چون n بازیکن داریم که همه با هم بازی می‌کنند، تعداد مسابقات برابر است با $\frac{n(n-1)}{2}$.
- **گره‌های بازیکنان:** به ازای هر بازیکن یک گره می‌سازیم ($P_1, P_2, \ldots, P_n$).

یال‌ها و ظرفیت‌ها (Edges and Capacities):

- **از S به همه مسابقات (M_{ij}):** یک لوله می‌کشیم با ظرفیت 2. چرا؟ چون در هر مسابقه دقیقا 2 امتیاز برای توزیع وجود دارد.
- **از هر مسابقه (M_{ij}) به دو بازیکن آن مسابقه (P_i و P_j):** دو لوله می‌کشیم، هر کدام با ظرفیت 2. چرا؟ چون یک بازیکن در یک مسابقه حداکثر می‌تواند 2 امتیاز بگیرد (در صورت برد).
- **از هر بازیکن (P_i) به مقصد (T):** یک لوله می‌کشیم با ظرفیتی برابر با امتیاز نهایی آن بازیکن (که کاسپاروف نوشته است). این یعنی لوله متصل به بازیکن اول، دقیقا به اندازه امتیاز ادعا شده‌ی او کشش دارد.

لم ۱. اگر در یک شبکه شار (*Flow Network*)، ظرفیت (*Capacity*) تمام یال‌ها اعداد صحیح *(Natural numbers/Integers)* باشند، آنگاه یک شار بیشینه وجود دارد که مقدار شار عبوری از هر یال آن نیز یک عدد صحیح است.

اثبات. الگوریتم فورد-فالکرسون برای یافتن شار بیشینه به این صورت عمل می‌کند:
۱. شروع کار: ابتدا شار تمام یال‌ها را صفر قرار می‌دهیم ($f(u, v) = 0$ برای تمام یال‌ها). عدد صفر یک عدد صحیح است.
۲. شبکه باقی‌مانده (Residual Graph): در هر مرحله، بر اساس شار فعلی، یک شبکه باقی‌مانده (G_f) می‌سازیم.
ظرفیت‌های باقی‌مانده در این شبکه طبق فرمول زیر محاسبه می‌شوند:
$c_f(u, v) = c(u, v) - f(u, v)$ (ظرفیت یال‌های مستقیم)
$c_f(v, u) = f(u, v)$ (ظرفیت یال‌های معکوس)
چون در شروع کار c و f هر دو صحیح هستند، تمام ظرفیت‌های باقی‌مانده (c_f) نیز صحیح خواهند بود.

۳. یافتن مسیر افزایشی (Augmenting Path): اگر مسیری از مبدأ (s) به مقصد (t) در شبکه باقی‌مانده وجود داشته باشد، آن را انتخاب می‌کنیم.
۴. تعیین مقدار افزایش (Δ): مقدار افزایشی که می‌توانیم به شار اضافه کنیم، برابر است با حداقل ظرفیت یال‌های موجود در مسیر افزایشی در شبکه باقی‌مانده:

$$\Delta = \min\{c_f(u, v) : (u, v) \in \text{path}\}$$

چون تمام ظرفیت‌های شبکه باقی‌مانده در این مرحله صحیح بودند، مقدار Δ نیز قطعاً یک عدد صحیح مثبت خواهد بود.
۵. به‌روزرسانی شار: شار جدید با اضافه کردن یا کم کردن Δ از شار قبلی به دست می‌آید. از آنجا که شار قبلی صحیح بود و Δ هم صحیح است، شار جدید نیز صحیح باقی می‌ماند.
۶. تکرار: این فرایند تا زمانی که دیگر مسیر افزایشی پیدا نشود ادامه می‌یابد.

از آنجا که الگوریتم با شار صحیح (صفر) شروع می‌شود و در هر گام یک مقدار صحیح به شار یال‌ها اضافه می‌کند، وقتی الگوریتم متوقف می‌شود (که در ظرفیت‌های صحیح حتماً متوقف می‌شود)، شار نهایی که همان شار بیشینه است، در تمام یال‌ها مقداری صحیح خواهد داشت. □

مراحل دقیق و گام‌به‌گام الگوریتم

گام اول: بررسی اولیه (حالت‌های بدیهی)

می‌دانیم در یک تورنمنت دوره‌ای با n بازیکن، تعداد کل مسابقات برابر با $\frac{n(n-1)}{2}$ است. در هر مسابقه دقیقا 2 امتیاز توزیع می‌شود. پس مجموع کل امتیازات تمام بازیکنان باید دقیقا برابر با $n(n-1)$ باشد. فرض کنیم امتیاز ادعا شده برای بازیکن iام را با s_i نشان دهیم. اگر $\sum_{i=1}^{n} s_i \neq n(n-1)$ باشد، جدول قطعا نامعتبر است و الگوریتم متوقف می‌شود. در غیر این صورت، شبکه زیر را می‌سازیم.

گام دوم: ساخت گراف شبکه $G = (V, E)$

شبکه ما شامل مجموعه گره‌ها و یال‌های جهت‌دار زیر است:

- یک گره مبدا (S) و یک گره مقصد (T).
- **گره‌های مسابقات:** به ازای هر جفت بازیکن i و j (که $1 \leq i < j \leq n$)، یک گره به نام M_{ij} ایجاد می‌کنیم.
- **گره‌های بازیکنان:** به ازای هر بازیکن، یک گره به نام P_i ایجاد می‌کنیم ($1 \leq i \leq n$).
- از S به هر گره M_{ij} یک یال با ظرفیت $c(S, M_{ij}) = 2$ رسم می‌کنیم.
- از هر گره مسابقه M_{ij}، دو یال به P_i و P_j با ظرفیت‌های $c(M_{ij}, P_i) = 2$ و $c(M_{ij}, P_j) = 2$ رسم می‌کنیم.
- از هر گره بازیکن P_i به گره مقصد T یک یال با ظرفیت $c(P_i, T) = s_i$ رسم می‌کنیم.

گام سوم: اجرای الگوریتم

الگوریتم یافتن بیشینه شار (Maximum Flow) مانند «ادمونز-کارپ» (Edmonds-Karp) یا «فورد-فالکرسون» را روی این شبکه از مبدا S به مقصد T اجرا می‌کنیم تا مقدار شار بیشینه به دست آید.

گام چهارم: بررسی صحت جدول

پس از اجرای الگوریتم، اگر مقدار شار بیشینه دقیقا برابر با مجموع امتیازات کل (یعنی $n(n-1)$) باشد، جدول امتیازات معتبر است. در غیر این صورت، جدول نامعتبر و ساختگی است.

گام پنجم: استخراج ریز نتایج بازی‌ها

در صورتی که جدول معتبر بود، برای فهمیدن نتیجه بازی بین بازیکن i و j، به میزان شار عبوری (Flow) از گره M_{ij} به گره‌های P_i و P_j نگاه می‌کنیم (آن را با f نشان می‌دهیم):

- اگر $f(M_{ij}, P_i) = 2$ و $f(M_{ij}, P_j) = 0$: بازیکن i برنده و بازیکن j بازنده شده است.
- اگر $f(M_{ij}, P_i) = 1$ و $f(M_{ij}, P_j) = 1$: بازی مساوی شده است.
- اگر $f(M_{ij}, P_i) = 0$ و $f(M_{ij}, P_j) = 2$: بازیکن i بازنده و بازیکن j برنده شده است.

(نکته: چون ظرفیت‌ها اعداد صحیح هستند، الگوریتم‌های شار نیز مقادیر صحیح برای شار یال‌ها برمی‌گردانند، لذا مقادیر کسری نخواهیم داشت). $\leftarrow$ اثبات در لم اول

گام ششم: تحلیل پیچیدگی زمانی

تعداد گره‌های گراف ما $O(n^2)$ و تعداد یال‌های آن نیز $O(n^2)$ است. از آنجا که الگوریتم‌های استاندارد شار شبکه (مانند ادمونز-کارپ) دارای پیچیدگی زمانی چندجمله‌ای نسبت به تعداد گره‌ها و یال‌ها هستند ($O(VE^2)$)، کل این فرآیند در زمان چندجمله‌ای (Polynomial Time) اجرا می‌شود و محدودیت مسئله را برآورده می‌کند.
