طراحی الگوریتم

پاسخنامه تمرین ششم - NP

آوا میرمحمدمهدی

در حل سوالات می‌توانید از NP-Complete بودن مسائل Independent Set و Vertex Cover علاوه بر مسائل اسلایدهای درس استفاده کنید.

۱. بازی باینری — ۱۵ نمره

دو مساله‌ی A و B را در نظر بگیرید که هر کدام امکان برنده شدن یا نشدن در بازی گفته شده را تصمیم‌گیری می‌کنند. A را در زمان چندجمله‌ای به B کاهش دهید.

A: در این بازی تعدادی مهره داریم که پشت و روی آنها با رنگ‌های قرمز و آبی مشخص شده است و بر روی هر طرف آن کلمه‌ای با استفاده از الفبای انگلیسی نوشته شده است. (کلمه می‌تواند شامل یک حرف یا تعدادی حروف باشد و کلماتی که در پشت و روی یک مهره نوشته شده‌اند لزوما باهم یکسان نیستند.) همچنین از هر نوع مهره، هر تعداد که بخواهیم موجود است و اجازه داریم در زمان انتخاب مهره، پشت و روی آن را ببینیم. برنده در این بازی فردی است که بتواند تعداد متناهی از این مهره‌ها را در کنار هم قرار دهد به طوری که دنباله حروفی که از کنار هم قرار گرفتن لغات پشت مهره‌ها تشکیل می‌شود با دنباله حروفی که از کنار هم قرار گرفتن لغات روی مهره‌ها ایجاد می‌شود برابر باشد. واضح است تضمین نمی‌شود که برنده شدن در تمامی حالات ممکن باشد.

برای مثال اگر در $\frac{ab}{a}$ کلمه بالایی را کلمه‌ای که در روی مهره و کلمه‌ی پایینی را کلمه‌ای که در پشت مهره نوشته شده در نظر بگیریم، با چیدن مهره‌ها به صورت زیر برنده بازی خواهیم شد:

$$\frac{a}{ab}, \frac{bc}{ca}, \frac{a}{ab}, \frac{abc}{c}$$

B: این بازی دقیقا مانند بازی A است با این تفاوت که در تشکیل کلمات به جای حروف الفبای انگلیسی از اعداد باینری استفاده می‌شود.

برای مثال با چیدن مهره‌ها به صورت زیر برنده بازی خواهیم شد:

$$\frac{01}{011}, \frac{11}{10}, \frac{00}{01}, \frac{11}{1}$$

پاسخ :

فرض می‌کنیم الفبای انگلیسی برابر با $\Sigma = \{c_1, c_2, \ldots, c_n\}$ باشد به طوری که c_i نشان دهنده‌ی یکی از کاراکترهای الفباست. برای تبدیل این ورودی به ورودی مساله‌ی B، کافیست هر کاراکتر c_i را با $011\ldots11$ نمایش دهیم به طوری که تعداد 1 ها برابر با i تا باشد. برای مثال دومینوی $\frac{c_1c_3}{c_5}$ با $\frac{010111}{011111}$ جایگزین خواهد شد. واضح است که مسابقه A دارای برنده خواهد بود اگر وتنها اگر مسابقه B دارای برنده باشد زیرا 0 به عنوان جداکننده و مشخص کننده تفاوت بین حروف الفبا عمل می‌کند و آنها را به عدد باینری تبدیل می‌کند در نتیجه دو طرف اثبات بدیهی خواهد بود.

۲. رئوس تنها ۱۵ نمره

زیرمجموعه L از رئوس در یک گراف غیرجهت‌دار را «تنها» می‌نامیم اگر هر راس در L حداکثر با یک راس دیگر در L مجاور باشد. ثابت کنید مساله‌ی مشخص کردن اینکه یک گراف، زیرمجموعه‌ای «تنها» به اندازه k دارد یا خیر، متعلق به NP-Hard است.

پاسخ :

برای اثبات، مساله‌ی Independent-Set که یک مساله‌ی NP-Complete است را به مساله داده شده کاهش می‌دهیم. فرض کنیم ورودی مساله‌ی Independent-Set، $< G, k >$ و ورودی مساله‌ی صورت سوال $< G', k' >$ باشند، برای ساخت G' از روی G، ابتدا به ازای هر راس v_i که در G وجود دارد، یک راس v'_i اضافه می‌کنیم و یال $v_iv'_i$ را تشکیل می‌دهیم؛ پس مجموعه یال‌های گراف G' به صورت $E' = \{v_iv'_i | \forall v_i \in G\} \cup E$ خواهد بود. همچنین اگر تعداد رئوس گراف G را برابر با n در نظر بگیریم، $k' = k + n$ خواهد بود. حال ثابت می‌کنیم Independent-Set، $< G, k >$ را می‌پذیرد اگر وتنها اگر مساله‌ی داده شده $< G', k' >$ را بپذیرد. برای طرف اول اثبات، واضح است که اگر مجموعه L در گراف G مستقل باشد، مجموعه $L' = S \cup \{v'_1, v'_2, \ldots, v'_n\}$ در گراف G' «تنها» است. حال برای اثبات طرف دیگر، فرض کنید مجموعه L' در گراف G' «تنها» باشد؛ این مجموعه دارای $k + n$ راس است پس حداقل k جفت راس $\{v_i, v'_i\}$ در مجموعه L' وجود دارد. k راس بدون پریم که به صورت v_i هستند را جدا می‌کنیم و آنها را مجموعه L می‌نامیم؛ این مجموعه در گراف G، یک مجموعه «تنها» ایجاد می‌کند زیرا این رئوس در L' حداکثر یک راس مجاور با راس متناظر خود داشتند و در مجموعه L یالی نخواهند داشت.

۳. موش و پنیر ۱۵ نمره

موشی در یک شبکه‌ی گراف جهت‌دار گیر افتاده است و می‌خواهد هرچه سریعتر پنیرهایی که روی نودهای این گراف است را بخورد. خانه‌ی این موش روی گره s قرار دارد و گراف از L حلقه $C = \{R_1, R_2, \ldots, R_L\}$ تشکیل شده است به طوری که تمامی این حلقه‌ها از گره s شروع می‌شوند و در هر گره، یک تکه پنیر وجود دارد. می‌خواهیم بدانیم آیا موش می‌تواند با انتخاب یک زیرمجموعه k عضوی از C و پیمایش آنها، تمام پنیرهای موجود را بخورد؟ (این k حلقه ممکن است از یک گره بیش از یک بار عبور کنند). ثابت کنید که این مساله NP-Complete است.

پاسخ :

ابتدا ثابت می‌کنیم که مساله‌ی مذکور NP است؛ برای این کار باید نشان دهیم که verify کردن جواب، در زمان چندجمله‌ای قابل انجام است. بدین منظور کافیست دورهای انتخاب شده را پیمایش کنیم تا مطمئن شویم که از تمامی رئوس عبور کرده‌ایم. برای اثبات NP-Hard بودن مساله، مساله‌ی Vertex-Cover را به مساله‌ی مذکور کاهش می‌دهیم. فرض کنید ورودی مساله Vertex-Cover به صورت $< G = (V, E), k >$ باشد و $E(v_i)$ را برابر با تمام یال‌های متصل به گره v_i تعریف کنیم؛ حال ورودی مساله‌ی موش و پنیر که $< G' = (V', E'), C, k' >$ است را از روی ورودی Vertex-Cover به شکل زیر می‌سازیم:

$$V' = V \cup \{s\}$$

$$C_i = \{s, E(v_i), s\}$$

$$k' = k$$

حال باید ثابت کنیم مساله‌ی Vertex-Cover ورودی‌اش را می‌پذیرد اگر وتنها اگر مساله‌ی موش و پنیر بپذیرد. برای طرف اول اثبات می‌دانیم اگر k راس وجود داشته باشد که همه یال‌ها را پوشش دهد، به ازای هر راس یک دور در G' وجود دارد که راس‌های متناظر با همان یال‌ها را پوشش می‌دهد. از طرفی برای طرف دیگر اثبات، اگر k دور در G' وجود داشته باشد که همه‌ی رئوس را پوشش دهد، به ازای هر دور، راسی در G وجود دارد که یال‌های متناظر با همان راس‌ها را پوشش می‌دهد. همانطور که دیدیم با اثبات NP و NP-Hard بودن مساله، NP-Complete آن ثابت شد.

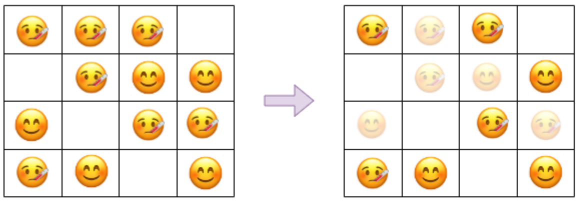
۴. بیماری بولا ۲۰ نمره

موج بیماری «بولا» از مهرماه امسال شروع شده و شیوع آن رو به افزایش است و متاسفانه تعداد زیادی از دانشجویان به آن مبتلا شده‌اند. کلاسی با n ردیف داریم به طوری که در هر ردیف m صندلی وجود دارد و دانشجویان مبتلا و سالم در برخی از این صندلی‌ها نشسته‌اند. می‌دانیم انتشار این ویروس عجیب در این کلاس در صورتی کنترل می‌شود که بتوان تعدادی از دانشجویان را به گونه‌ای از کلاس خارج کرد که دو شرط زیر برقرار باشد:

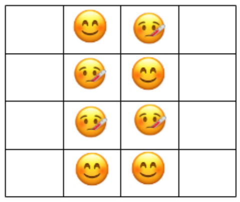
- در هر ردیف از کلاس حداقل یک نفر نشسته باشد. (چه فرد مبتلا و چه فرد سالم تفاوتی ندارد.)
- در هیچ ستونی دو نوع دانشجوی مبتلا و سالم وجود نداشته باشند.

ثابت کنید فهمیدن اینکه می‌توان بیماری بولا را با داشتن یک کلاس به همراه دانشجویانش کنترل کرد یا خیر، در دسته مسائل NP-Hard قرار دارد. (قطعا برای برخی حالات قرارگیری دانشجویان نمی‌توان این بیماری را کنترل کرد.)

برای مثال در شکل زیر در حالت (۱) با حذف دانشجویان نشان داده شده می‌توان بیماری را کنترل کرد ولی در حالت (۲) این کار امکان‌پذیر نیست.



(1)



(2)

پاسخ :

برای اثبات NP-Hard بودن مساله، کافی است مساله‌ی 3-SAT را به مساله‌ی داده شده کاهش دهیم. فرض کنید Φ یک 3CNF با m متغیر و n عبارت باشد؛ برای تبدیل این عبارت به ورودی مساله‌ی داده شده، برای هر صندلی موجود در ردیف i و ستون j به صورت زیر عمل می‌کنیم:

- اگر متغیر a_j در عبارت i ام از Φ وجود داشته باشد، فرد سالم را در صندلی (i, j) که در ردیف i و ستون j قرار دارد، می‌نشانیم.
- اگر متغیر $\overline{a_j}$ در عبارت i ام از Φ وجود داشته باشد، فرد مبتلا را در صندلی (i, j) که در ردیف i و ستون j قرار دارد، می‌نشانیم.
- اگر دو حالت بالا نبود، در صندلی (i, j) هیچ فردی را نمی‌نشانیم.

ثابت می‌کنیم که می‌توان به حالتی رسید که بیماری بولا کنترل شود اگر وتنها اگر Φ دارای جواب باشد. اگر Φ دارای جواب باشد، دو حالت برای متغیر a_j وجود دارد: اگر $a_j = True$ باشد، آنگاه تمام افراد مبتلا که در ستون j هستند را خارج می‌کنیم و اگر $a_j = False$ باشد، آنگاه تمام افراد سالم که در ستون j هستند را خارج می‌کنیم. هر متغیر در حداقل یک عبارت حضور دارد پس در هر ستون حداکثر یکی از انواع افراد (سالم یا مبتلا) قرار دارد و همچنین به دلیل اینکه هر عبارت حداقل یک متغیر که مقدار آن $True$ باشد وجود دارد، در هر ردیف حداقل یک دانشجو (مبتلا یا سالم) قرار گرفته است. برای طرف دوم اثبات اگر بتوانیم دانشجویان را به گونه‌ای خارج کنیم که بیماری کنترل شود، مقدار متغیر a_j مطابق زیر تعیین می‌شود:

- اگر دانشجویی سالم در ستون j قرار گرفته باشد، $a_j = True$ خواهد بود.
- اگر دانشجویی مبتلا در ستون j قرار گرفته باشد، $a_j = False$ خواهد بود.

۵. زیرگراف کامل   ۲۰ نمره

ثابت کنید اگر G یک گراف بدون جهت باشد، تعیین اینکه دارای یک زیرگراف کامل با حداقل $\lceil \frac{m}{2} \rceil$ گره است یا خیر، یک مساله‌ی NP-Complete است. (m تعداد رئوس گراف G است.)

پاسخ :

مساله‌ی داده شده را H-Clique می‌نامیم. ابتدا باید ثابت کنیم که این مساله در کلاس NP قرار دارد. برای این کار verifier ما، رئوسی که در clique وجود دارند را در نظر می‌گیرد و ابتدا تعداد رئوس آن را با تعداد رئوس گراف مقایسه می‌کند تا کمتر از $\frac{m}{2}$ نباشد و سپس به ازای هر دو راس u و v که در گواهی وجود دارند بررسی می‌کند که (u, v) در گراف وجود داشته باشد. بدیهی است که بررسی تمام این شروط در زمان چندجمله‌ای امکان‌پذیر است پس می‌توان گفت که این مساله در دسته NP قرار دارد.

حال برای اینکه اثبات کنیم این مساله در دسته NP-Hard قرار دارد، مساله‌ی Clique را که در دسته‌ی NP-Complete است به آن کاهش می‌دهیم. فرض کنید ورودی مساله Clique را به صورت $< G, k >$ نشان دهیم به طوری که k حداقل تعداد رئوس Clique باشد. اگر تعداد رئوس گراف G را برابر با m در نظر بگیریم، برای تشکیل گراف H که ورودی مساله‌ی H-Clique است سه حالت ممکن است رخ دهد:

۱) $k = \frac{m}{2}$: در این صورت گراف H را دقیقا همان گراف G در نظر می‌گیریم.

۲) $k > \frac{m}{2}$: در این حالت تعداد $2k - m$ راس با درجه 0 به گراف اضافه می‌کنیم که با این کار تعداد کل رئوس گراف برابر با $2k$ خواهد شد. گراف جدید همان گراف H خواهد بود.

۳) $k < \frac{m}{2}$: اگر این حالت رخ دهد، باید تعداد رئوس گراف و k را به طور همزمان افزایش دهیم. در این حالت t راس به گراف اضافه می‌کنیم و از این رئوس به تمام رئوس قبلی و همچنین به تمام رئوس جدید یال می‌دهیم. حال باید یک clique با اندازه $k + t$ در گراف جدید (H) پیدا کنیم. در نتیجه تعداد رئوس اضافه شده از رابطه $k + t = \frac{m+t}{2} \rightarrow t = m - 2k$ بدست می‌آید.

در هر کدام از حالات ذکر شده می‌توان گفت در گراف G یک clique با حداقل اندازه k وجود دارد اگر وتنها اگر مساله‌ی H-Clique برای گراف H دارای پاسخ باشد. برای تبدیل پاسخ‌های دو مساله به یکدیگر، هر کدام از حالات را جداگانه بررسی می‌کنیم:

۱) $k = \frac{m}{2}$: در این حالت گراف‌های H و G با هم یکسان هستند و clique پیدا شده در هر یک از مسائل، دقیقا پاسخ مساله دیگر است.

۲) $k > \frac{m}{2}$: در این حالت هم رئوس اضافه شده در گراف H عضو clique نخواهند بود و پاسخ پیدا شده برای هر یک از مسائل، پاسخ قابل قبولی برای مساله دیگر نیز هست.

۳) $k < \frac{m}{2}$: در این حالت ممکن است تعدادی از t راس اضافه شده در پاسخ مساله H-Clique قرار داشته باشند. اگر A یک پاسخ برای مساله‌ی H-Clique باشد، می‌دانیم $|A| \geq t + k$ است. با توجه به اینکه با اضافه کردن دقیقا t راس به گراف G، گراف H ساخته شده است، حداقل k راس از رئوس A باید از گراف قدیمی انتخاب شده باشند. پس اگر مجموعه رئوس اضافه شده را T در نظر بگیریم، $B = A/T$ یک پاسخ برای مساله Clique است. $(|B| \geq k)$.

اگر B را یک پاسخ برای مساله‌ی Clique با حداقل اندازه k در نظر بگیریم، آنگاه $A = B \cup T$ یک پاسخ با حداقل اندازه $t + k$ (نصف تعداد رئوس گراف H) براس مساله H-Clique خواهد بود.

تبدیل ذکر شده در زمان چندجمله‌ای قابل انجام است پس می‌توان گفت مساله H-Clique در دسته‌ی NP-Hard قرار دارد. همچنین پیش‌تر اثبات شده بود که NP هم هست پس در کلاس NP-Complete قرار دارد.

۶. درست یا نادرست ۲۰ نمره

فرض کنید مساله A در دسته‌ی NP-Complete قرار دارد و در زمان چندجمله‌ای قابل کاهش به مساله‌ی B است؛ همچنین می‌توان مساله‌ی C را در زمان چندجمله‌ای به A کاهش داد. درستی یا نادرستی عبارات زیر را با ذکر دلیل مشخص کنید.

الف) اگر بتوان مساله‌ی C را به طور قطعی در زمان چندجمله‌ای حل کرد، آنگاه $P = NP$ خواهد بود.

ب) اگر در آینده ثابت شود که نمی‌توان مساله‌ی B را با الگوریتمی چندجمله‌ای حل کرد، آنگاه ثابت می‌شود که $P \neq NP$ خواهد بود.

ج) اگر اثبات شود که $P \neq NP$ آنگاه قطعا نمی‌توان مساله‌ی A را در زمان چندجمله‌ای حل کرد.

د) اگر یک راه حل با پیچیدگی زمانی $\mathcal{O}(n^2)$ برای مساله‌ی B وجود داشته باشد، آنگاه می‌توان مساله‌ی A را نیز با الگوریتمی با پیچیدگی زمانی $\mathcal{O}(n^2)$ حل کرد.

پاسخ :

الف) نادرست - همانطور که می‌دانیم $P \subseteq NP$ پس C می‌تواند متعلق به دسته‌ی P باشد.

ب) نادرست - مساله‌ی B لزوما در دسته‌ی NP قرار ندارد پس نمی‌توان چنین نتیجه‌ای گرفت.

ج) درست - در صورتی که $P \neq NP$ باشد، هیچ مساله‌ای که عضو NP-Complete باشد را نمی‌توان در زمان چندجمله‌ای حل کرد زیرا اگر مساله‌ای در این دسته وجود داشته باشد که در زمان چندجمله‌ای قابل حل باشد، تمام مسائل دسته‌ی NP در زمان چندجمله‌ای حل خواهند شد و $P = NP$ خواهد بود.

د) نادرست - کاهش A به B در زمان چندجمله‌ای انجام می‌شود پس A در زمان چندجمله‌ای حل خواهد شد ولی ممکن است پیچیدگی آن بیشتر از $\mathcal{O}(n^2)$ باشد.