بسم‌الله الرحمن الرحیم

طراحی الگوریتم ـ بهار ۱۴۰۰

NP

امیرحسین عباسکوهی

تاریخ تحویل : یکشنبه (۱۴۰۰/۰۳/۳۰)

نکته: در مسائل NP-complete باید حتما دو بخش را اثبات کنید. اول اینکه مسئله داده شده NP است. سپس ثابت کنید مسئله NP-complete شناخته شده ای با پیچیدگی زمانی چند جمله ای وجود دارد که می‌توان به مسئله داده شده کاهش داد.

۱. درباره مسائل NP به سوالات زیر پاسخ دهید.(۱۰ نمره)

- تعریفی از مسائل NP ارائه دهید.
- تعریفی برای مسائل NP-Hard ارائه دهید.
- تعریفی برای مسائل NP-Complete ارائه دهید.
- چرا از reduction در حل مسائل NP-Complete استفاده می‌شود؟

(پاسخ)

- مسائل NP : مسایلی که خودش شاید در زمان چندجمله‌ای حل نشود، ولی اگر یک راه‌حلش را داشته باشیم، می‌توانیم درستی آن‌را در زمان چندجمله‌ای وارسی کنیم، NP است.
- مسائل :NP-Complete دسته از مسائل که در صورت داشتن یک مدرک می‌توان درستی آن را در زمان چند جمله ای بررسی کرد. در صورت حل شدن این مسائل تمامی مسائل NP حل خواهند شد. این مسائل حتما در دسته NP قرار دارند.
- مسائل :NP-Hard این مسائل الزاما در کلاس NP قرار ندارند یعنی نمی توان در زمان چند جمله درستی آن ها را با داشتن یک مدرک بررسی نمود. مسائل NP-Complete زیر مجموعه مسائل NP-Hard می باشند.
- یکی از شرایط NP-Comeplete بودن این است که ثابت کنیم تمام مسائل NP قابل حل توسط NP-Complete ها می‌باشد. پس نیاز به کاهش داریم.

۲. درستی یا نادرستی عبارت های زیر مشخص کنید و دلیل خود را در این باره ذکر کنید.(۱۰ نمره)

(آ) اگر $X \leq_P Y$ و $Y \leq_P Z$ آنگاه $X \leq_P Z$

(ب) اگر $X \leq_P Y$ و Y یک مسئله NP-hard باشد آنگاه X نیز NP-hard می‌باشد.

(ج) فرض کنید X یک مسئله در کلاس NP باشد و $P \neq NP$ آنگاه X نمی‌تواند در زمان چند جمله ای حل شود.

(د) اگر یک مسئله NP-complete در زمان خطی حل شود، تمامی مسائل NP-complete را می‌توان در زمان خطی حل کرد.

(پاسخ)

(آ) درست است. فرض کنید Z در زمان $T(|Z|)$ حل شود. چون $Y \leq_P Z$ دو چند جمله ای r و s وجود دارد که می‌توان Y را در زمان $r(|Y|) + s(|Y|)T(|Y|)$ حل کرد. همین منطق برای X هم صادق است پس X را می‌توان در زمان $p(|X|) + q(|X|)r(|X|) + s(|X|)T(|X|) = p'(|X|) + q'(|X|)T(|X|)$حل کرد که در آن p' و q' چند جمله ای هستند. پس $X \leq_P Z$.

(ب) نادرست است. رابطه داده شده نشان دهنده آن است که Y مسئله سخت تری نسبت به X می‌باشد و نه برعکس، یعنی به طور پیش فرض X قابل کاهش به هر مسئله ای مانند Y است اما NP-hard نیست.

(ج) نادرست است زیرا $\emptyset \neq P \subseteq NP$ و مسائل P در زمان چند جمله ای قابل حل شدن هستند پس X می‌تواند عضو P هم باشد.

(د) نادرست است. اگر P برابر NP بود می‌شد چنین ادعایی داشت اما از آنجا که نمی‌توان چنین ادعایی بدون اثبات کرد پس این گزاره نادرست است.

۳. در مسئله پیدا کردن Clique (در این مسئله به دنبال زیر مجموعه ای شامل K راس هستیم که آن رئوس همگی با هم مجاور اند یعنی دو به دو به هم یال دارند) ،طبیعی است که به دنبال راس های با درجه بالا بگردیم. فرض کنید میخواهیم بررسی کنیم که آیا گراف داده شده دارای یک Clique با درجه رئوس پایین هست یا نه(به طور مشخص کوچکتر یا مساوی میانه درجه رئوس کل گراف). در اینجا میخواهیم نشان دهیم این مسئله هم NP-complete می باشد.(۲۵ نمره)

کلیک درجه پایین(Low-Degree Clique): گراف $G = (V, E)$ و عدد صحیح k به ما داده شده است. آیا گراف G کلیکی به اندازه k شامل رئوسی که درجه آن ها بزرگتر از میانه درجه رئوس کل گراف نباشد، دارد؟

(پاسخ)

برای اینکه ثابت شود مسئله LDC یک مسئله NP-complete است با دو مرحله زیر را انجام دهیم:

ابتدا ثابت می‌کنیم که $LDC \in NP$. برای اینکار k راس از G را به عنوان پاسخ حدس میزنیم. حال درجه تمام رئوس G را محاسبه می‌کنیم و در نهایت میانه آن‌ها را به دست می آوریم. اینکار میتواند در زمان $O(n+m)$ به وسیله الگوریتم انتخاب سریع [1] انجام شود. حال بررسی می‌کنیم که آیا همه رئوس حدس زده شده، درجه کوچکتر مساوی با میانه دارند یا خیر. همچنین بررسی می‌کنیم هر کدام از این رئوس با بقیه رئوس همسایه باشد. اگر هر دو شرط یاد شده برقرار باشند، خروجی "بله" خواهد بود و در غیر این صورت پاسخ "خیر" خواهد بود. اگر گراف G دارای LDC باشد در این صورت باید یکی از حدس های ما پاسخ "بله" دریافت کند.

حال ثابت می‌کنیم که LDC یک مسئله NP-hard است. به این منظور، نشان می‌دهیم که مسئله استاندارد کلیک(Standard Clique) به صورت چند جمله ای قابل کاهش به LDC می‌باشد($LDC \leq_P LDC$). ایده این است که به صورت مصنوعی درجه میانه گراف G را افزایش دهیم تا هر کلیک در گراف G یک LDC در گراف جدید باشد.

فرض کنید به ما یک گراف G و یک عدد صحیح k به عنوان ورودی داده شده است که n تعداد رئوس در G است و $k \geq ۳$. (در غیر اینصورت مسئله به تشخیص اینکه آیا G حداقل یک یال دارد کاهش می‌یابد.) ما گراف جدید G' با استفاده از یک کپی از G می‌سازیم و به آن یک گراف کامل دو بخشی $n \times n$ اضافه می‌کنیم. گراف به دست آمده گرافی با $۳n$ راس می‌باشد. راس هایی که از G به G' رسیده‌اند حداکثر درجه $n - ۱$ دارند و راس های اضافه شده هم همگی درجه n دارند بنابراین میانه درجه رئوس در گراف G' برابر با n خواهد شد. پس توانستیم گراف مد نظر خود را ایجاد کنیم و این کار در زمان چند جمله ای انجام شد.

یعنی G دارای کلیک به اندازه k است اگر و تنها اگر G' یک LDC به اندازه k داشته باشد.

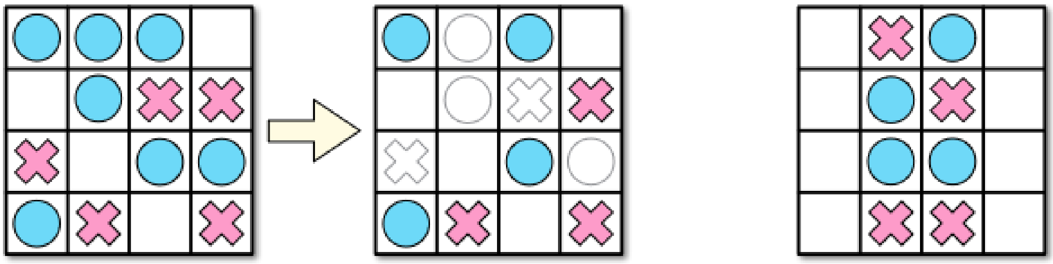
۴. جدولی با ابعاد $n \times m$ از مربع های واحد داریم که هر کدام از این مربع ها می‌تواند خالی، دارای X و یا دارای O باشد. هدف این است تا با حذف برخی از X یا O ها جدول را به شکلی در آوریم که دارای دو شرط زیر باشد:

اول– هر ردیف حداقل دارای یکی از علامت های X یا O باشد.
دوم– هیچ ستونی دارای هر دو علامت نباشد.

قطعا برای برخی حالات نمی‌توان شرایط بالا را به دست آورد.

ثابت کنید فهمیدن اینکه برای یک جدول اولیه فهمیدن اینکه می‌توان به جدول با شرایط ذکر شده رسید یا خیر یک مسئله NP-hard می‌باشد.(۲۰ نمره)

[1] Quik selection algorithm



A solvable puzzle and one of its many solutions. An unsolvable puzzle.

(پاسخ)

با کاهش مسئله از $3SAT$ به مسئله داده شدهٔ ثابت می‌کنیم مسئله NP-hard می باشد.

فرض کنید Φ یک $3CNF$ با m متغیر و n عبارت باشد. ما این عبارت را تبدیل به توصیفی از جدولی خوئمان می کنیم:

می دانیم که اندازه جدول برابر است $n \times m$. حال برای هر خانه جدول در سطر i و ستون j داریم:

- اگر متغیر x_j در عبارت iام از Φ وجود داشته باشد علامت X را در خانه (i, j) قرارا می‌دهیم.
- اگر متغیر $\bar{x_j}$ در عبارت iام از Φ وجود داشته باشد علامت O را در خانه (i, j) قرارا می‌دهیم.
- در غیر این صورت خانه (i, j) را خالی قرار می‌دهیم.

نتیجه می‌گیریم که جدول پاسخ دارد اگر و تنها اگر Φ پاسخ داشته باشد(satisfiable باشد).

در اینجا اگر Φ پاسخ داشته باشد دو حالت برای متغیر x_j داریم: اگر $x_j = True$ باشد آنگاه تمام علامت های O را از جدول حذف می‌کنیم و اگر $x_j = False$ باشد تمام علامت های X را از جدول حذف می‌کنیم. چون هر متغیر در حداقل یک عبارت حضور دارد پس هر ستون حداکثر یک نوع از علامت ها را در خود دارد. از طرف دیگر چون هر عبارت حداق یک بخش درست دارد پس هر سطر حداقل یک علامت را دارد.

از طرف دیگر اگر جدول پاسخ داشته باشد مقدار متغیر x_j از طریق روش زیر تعیین می‌شود:

- اگر ستون j شامل علامت X باشد آنگاه $x_j = True$.
- اگر ستون j شامل علامت O باشد آنگاه $x_j = False$.
- در غیر این صورت x_j مقدار دلخواه می باشد.

۵. در مسئله دسته بندی، به ما مجموعه $A = \{a_1, a_2, ..., a_n\}$ داده شده است که a_i در این مجموعه یک عدد صحیح می باشد. از ما خواسته شده است که آیا $B \subseteq A$ ای وجود دارد که در آن $\sum\limits_{a_i \in A/B} a_i = \sum\limits_{a_i \in B} a_i$. نشان دهید که مسئله دسته بندی $NP-complete$ می‌باشد.(۲۰ نمره)

(پاسخ)

مسئله دسته بندی NP می باشد زیرا با داشتن یک پاسخ B می‌توان در زمان خطی مجموعه A/B را به دست آورد و نیز در زمان خطی $\sum\limits_{a_i \in A/B} a_i = \sum\limits_{a_i \in B} a_i$ را بررسی کنیم.

برای اثبات $NP-hard$ بودن، ما مسئله را از $Subset-Sum$ کاهش می‌دهیم. در مسئله $Subset-Sum$ به ما مجموعه $X = \{x_1, x_2, ..., x_n\}$ از اعداد نا منفی و همچنین یک هدف t داده شده است و از ما خواسته شده تا مجموعه $Y \subseteq X$ را پیدا کنیم به طوری که $\sum\limits_{x_i \in Y} x_i = t$. برای کاهش از $A = X \cup \{s+t, 2s-t\}$ استفاده می‌کنیم که در آن $s = \sum_{i=1}^{n} x_i$.

در ابتدا فرض کنید Y برای مسئله $Subset-Sum$ بر روی مجموعه X با هدف t می باشد. حال پاسخ B برای مسئله دسته بندی بر روی مجموعه A وجود دارد که برابر است با $S \cup \{2s-t\}$، زیرا مجموعه اعضای این مجموعه برای $2s$ است و مجموع اعضای $X/S \cup \{2s-t\}$ برابر است با: $(s-t)+(s+t) = 2s$.

حال فرض کنید B پاسخی برای مسئله دسته بندی بر روی مجموعه A باشد. در این صورت می‌توانیم پاسخی بر روی مجموعه X با هدف t پیدا کنیم. برای اینکار در نظر بگیرید که مجموع تمام اعضای A برابر است با $s + (s+t) + (2s-t) = 4s$، پس مجموع اعضای مجموعه B باید برابر $2s$ باشد. بنابراین دو عضوی که ما به A اضافه کردیم باعث اضافه شدن مقدار $3s$ به مجموع اعضای مجموعه A شده است پس باید یکی از آن ها در B و دیگری در A/B باشد. بدون تغییر در کلیت مسئله فرض کنید $2s - t \in B$ باشد. پس مجموع بقیه اعضای این مجموعه باید برابر t باشد. بنابراین این اعضا همان پاسخ مسئله $Subset - Sum$ بر روی مجموعه X می باشد.

با توجه به این تمام مراحل کاهش خطی بود پس مسئله $NP - hard$ است و با توجه به NP بودن آن مسئله دسته بندی $NP - complete$ می‌باشد.

۶. در مورد الگوریتم های تقریبی (approximation) جستجو کنید و آن را توضیح دهید. سپس یکی از الگوریتم های تقریبی معروف علوم کامپیوتر را بنویسید.(۱۵ نمره)

برای مطالعه می توانید به لینک های زیر رجوع کنید:

Approximation algorithm On Yale University

Approximation algorithm On Wikipedia