دانشگاه علم و صنعت ایران
تحلیل و طراحی الگوریتم ـ نیمسال دوم ۹۷ـ۹۸

امتحان پایان‌ترم ـ ۹۸/۴/۵
مدرس: سید صالح اعتمادی

۱. [۲۴نمره] کلاس‌های پیچیدگی:

(آ) کلاس پیچیدگی NP را تعریف کنید. یک مثال بزنید.

دسته‌ای از مسائل که درستی جواب آن را می‌توان در زمان چند جمله‌ای راست‌آزمایی کرد. مثال: پیدا کردن دور همیلتونی در گراف.

(ب) کلاس پیچیدگی P را تعریف کنید. یک مثال بزنید.

مجموعه مسائلی که در زمان چند جمله‌ای قابل حل هستند. مانند پیدا کردن درخت پوشای کمینه [1] .

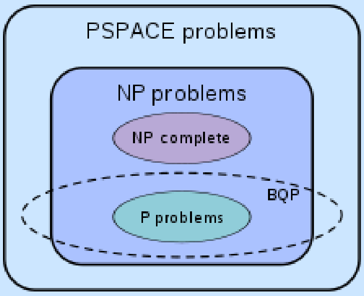
(ج) کلاس پیچیدگی NP-Complete را تعریف کنید. یک مثال بزنید.

زیرمجموعه‌ای از مسائل NP که تمام مسائل NP را می‌توان به آنها تقلیل [2] داد. مثال: مساله صدق پذیری یا SAT.

(د) کلاس پیچیدگی `bounded-error quantum polynomial time` را تعریف کنید. یک مثال بزنید.

زیر مجموعه‌ای از مسائل NP که با کوانتوم کامپیوتر در زمان چند جمله‌ای و با احتمال خطای «محدود» قابل حل می‌باشند. مانند پیدا کردن عوامل اول یک عدد که با الگوریتم کوانتومی Shor در زمان چند جمله‌ای قابل حل است.

(ه) نمودار ون [3] ارتباط کلاس‌های پیچیدگی بالا را بکشید.



منبع: ویکی‌پدیا

(و) مساله قرن P در برابر NP را توضیح دهید.

مساله این است که آیا کلاس‌های پیچیدگی P و NP برابرند. آیا مسائلی هستند که در زمان چند جمله‌ای قابل راست‌آزمایی هستند ولی در زمان چند جمله‌ای قابل حل نیستند؟

(ز) نتیجه عملی حل مساله P در برابر NP چیست؟

بستگی به جواب مساله دارد. اگر P برابر NP باشد. برای مساله‌های مشکلی مثل پیدا کردن دور همیلتونی راه حل چند جمله‌ای پیدا می‌شود. اگر برابر نباشند می‌توان از غیر قابل حل بودن بعضی مسائل در زمان چند جمله‌ای برای مسائل رمز نگاری با خیال راحت‌تر استفاده کرد. از این جهت که اثبات شده است که راه حل چندجمله‌ای برای این مسائل وجود ندارد.

(ح) اثبات کنید مساله صدق پذیری یا SAT عضو کلاس پیچیدگی NP-Complete است.

باید اثبات کنیم، تمام مسائل NP به SAT تقلیل پیدا می‌کنند.

مرحله اول اثبات: طبق تعریف جواب یک مساله NP (مثلا L) را می‌توان با یک الگوریتم چند جمله‌ای (G) راست‌آزمایی کرد. مدار الگوریتم G را طراحی می‌کنیم به طوریکه یک جواب A از ورودی دریافت کرده و اگر جواب درست باشد خروجی مدار 1 باشد اگرنه 0. بعد از ساخت مدار، می‌توانیم جواب مساله را با استفاده از مساله Circuit-SAT یا صدق پذیری مدار حل کنیم. یعنی جوابی را پیدا کنیم که خروجی مدار برای آن جواب یک باشد. در نتیجه تمام مسائل NP به Circuit-SAT تقلیل پیدا کردند.

مرحله دوم اثبات: می‌دانیم که Circuit-SAT به SAT تقلیل پذیر است.

در نتیجه تمام مسائل NP به SAT تقلیل پذیر هستند.

۲. [۴۸نمره] شما از جنگ و دعوا در جهان خسته شده‌اید و می‌خواهید قدمی در جهت ایجاد صلح جهانی بردارید. برای اینکار می‌خواهیم یک سازمان ملل جدید و واقعی ایجاد کنیم. چون سازمان ملل فعلی نماینده حکومت‌هاست. و حکومت با ملت فرق می‌کنه. از طرفی نماینده‌های حکومت‌ها هم به یکدیگر ربطی ندارند. از این جهته که معمولا کاری از پیش نمی‌برند. مثلا میگویند که این آقای جان کری چون دامادش ایرانی بود توانست به یک توافق تاریخی با ایران برسه. آقای ظریف هم که مدتی زیادی آمریکا تحصیل و زندگی کرده و حرف اونها رو خوب می‌فهمه. حالا فکر کنید اگر پسر آقای ظریف داماد جان کری بود، چی میشد؟ خوب

[3] نمودار مجموعه یا `Venn Diagram`

اصلا بهتر نبود همه نماینده‌ها یه نسبت فامیلی با نماینده‌های دیگه می‌داشتند؟ خوب این ایده سازمان ملل جدید ماست و شما مامور پیدا کردن نماینده‌های این «سازمان ملل» هستید [۴].

به صورت رسمی ملت‌ها/مردم را بر حسب کشور محل سکونت‌شان با یک گراف k بخشی [۵] $G(P_1, P_2, ..., P_k, E)$ بدون جهت و بدون وزن مدل کرده‌ایم. هر کشور به صورت یک بخش P_i، و مردم هر کشور به صورت گره‌های آن بخش مدل شده‌اند. طبق تعریف گراف k بخشی، بین گره‌های یک بخش یالی موجود نیست. دو فرد در دو کشور متفاوت تنها در صورتی با یک یال متصل می‌باشند که بین آنها رابطه فامیلی «قابل توجهی» موجود باشد. هدف انتخاب کردن یک گره از هر بخش است، به صورتی که تعداد یال‌های بین گره‌های انتخاب شده حداکثر باشد.

(آ) آیا این مساله NP است؟ چرا؟

خیر. چون مساله تصمیم‌گیری نبوده و راه حل چند جمله‌ای برای راست‌آزمایی آن نداریم.

(ب) یک راه حل این مساله بررسی تمام جواب‌های ممکن و انتخاب جواب بهینه است. تعداد جواب‌های ممکن را بر حسب پارامترهای مساله بیان کنید. جمعیت جهان را N، جمعیت هر کشور را n_i و تعداد کشورها را k فرض کنید. چنانچه نیاز به پارامتر دیگری است، آن را تعریف کرده و استفاده کنید.

توی هر کشور n_i نفر می‌توانند انتخاب بشوند. پس تعداد کل حالات می‌شود $\prod_{i=1}^{k} n_i$

(ج) با فرض تعداد ۲۰۰ کشور و متوسط یک میلیون جمعیت هر کشور، تعداد جواب‌های موجود چندتاست؟ درصورتی که بررسی هر جواب روی یک کامپیوتر ۱ میکروثانیه (10^{-6}) طول بکشد و ۱۰۰۰ کامپیوتر داشته باشیم، بررسی تمام جواب‌ها چقدر طول می‌کشد؟ (جواب تقریبی [۶] کافیست.)

$$(10^6)^{200} \times 10^{-6} \times 10^{-3} = 10^{1191}\text{seconds} = \frac{3 \times 10^{1191}}{10^8}\text{years} = 3 \times 10^{1183}\text{years}$$

جهت اطلاع و مقایسه، تخمین دانشمندان از عمر دنیا ۱۳/۷۹۹ میلیارد سال یا 13.799×10^9 سال است.

(د) آیا توانی بودن فضای جواب‌های ممکن، دلیل عدم وجود الگوریتم چند جمله‌ای برای پیدا کردن جواب بهینه است؟ اگر بله، چرا؟ اگر نه، مثال نقض بزنید.

خیر. مثلا تعداد درخت‌های پوشای یک گراف کامل برابر است با n^{n-2} ولی پیدا کردن درخت پوشای کمینه الگوریتم چند جمله‌ای دارد.

(ه) این مساله به صورت یک مساله بهینه‌سازی [۷] بیان شده است. صورت مساله تصمیم‌گیری [۸] آن را بیان کنید؟

یک گراف چند بخشی داریم. آیا می‌توان از هر بخش یک گره را انتخاب کرد به طوریکه تعداد کل یال‌های بین گره‌های انتخاب شده **حداقل** k باشد؟

(و) مساله بهینه‌سازی را به مساله تصمیم‌گیری معادل تقلیل [۹] دهید.

مساله تصمیم‌گیری را برای عدد k برابر یک حل می‌کنیم. اگر جواب مثبت بود k را دو برابر می‌کنیم. اگر جواب منفی بود بین مقدار فعلی و مقدار قبلی k به روش جستجوی دودویی بزرگترین k در مساله‌ای تصمیم‌گیری جواب مثبت دارد را پیدا می‌کنیم. طبق تعریف این جواب مساله‌ای بهینه سازی است.

(ز) برای این مساله تصمیم‌گیری راه حل چند جمله‌ای ارائه کرده یا اثبات کنید این مساله `NP-Complete` است.

پیدا کردن Clique با اندازه k در یک گراف با اندازه n از مسائل شناخته شده NPC [۱۰] است. در ادامه نشان می‌دهیم که حل کردن مساله تعریف شده بالا منجر به حل مساله Clique می‌شود. مساله Clique را برای یک گراف G در نظر بگیرید (شکل زیر را ببینید). یک گراف k بخشی R را از روی گراف G به این ترتیب می‌سازیم که در هر بخش از گراف جدید R، هر یک از n گره گراف G را تکرار می‌کنیم. مثلا به ازای گره u در G یک گره u_i در بخش iام اضافه می‌کنیم (به ازای i از ۱ تا k). سپس به ازای هر یال در G بین گره‌های u و v، تعداد $k-1$ یال در R بین گره‌های u_i و v_j ($i \neq j$) اضافه می‌کنیم. حالا مساله بالا را برای گراف k بخشی R حل کرده. در نتیجه گره‌هایی انتخاب می‌شوند که بیشترین تعداد یال بین آنها موجود باشد. براحتی می‌توان ملاحظه کرد که فقط

[۴] طراح این سوال هیچ تخصصی در علوم سیاسی ندارد و مسئولیتی در قبال درستی فرض‌های مساله قبول نمی‌کند.

[۵] `Multipartite or k-partite graph`

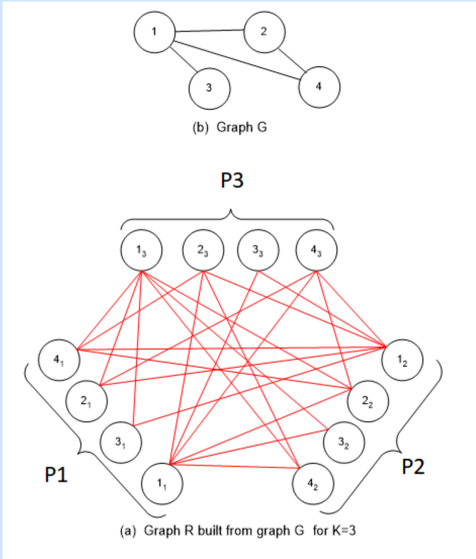
[۶] هر سه سال برابر ۹۴۶۰۸۰۰۰ ثانیه است. برای سادگی محاسبات هر سه سال را برابر 10^8 ثانیه بگیرید.

[۷] `Optimization Problem`

[۸] `Decision Problem`

[۹] `Reduce`

و فقط در صورت وجود Clique در گراف G، نتیجه حل مساله بالا یک Clique خواهد بود. بنابراین چنانچه تعداد یال‌های موجود بین گره‌های انتخاب شده در R برابر $\frac{k(k-1)}{2}$ باشد، جواب مساله Clique مثبت است و در غیر اینصورت منفی.
بنابراین نتیجه می‌گیریم می‌توان مساله Clique را با این مساله تقلیل داد و چون مساله Clique یک مساله NPC است، در نتیجه مساله بالا نیز عضو کلاس پیچیدگی NPC می‌باشد.



منبع: پایان‌نامه کارشناسی ارشد طراح سوال

(ح) در هر حال ممکن است نتوان مرتبه پیچیدگی محاسباتی مساله را بهتر کرد (چه الگوریتم چند جمله‌ای داشته باشیم، چه نداشته باشیم) ولی برای اعداد بزرگ تغییر ضریب ثابت [11] نیز می‌تواند تاثیر بسزایی داشته باشد. با فکر به این مساله در جهان واقعی ۳ ایده خود برای کوچکتر کردن این ضریب ثابت را بیان کنید.

برای حل مساله تصمیم‌گیری، که آیا می‌توان از هر بخش گره‌هایی انتخاب کرد که در نتیجه تعداد یال‌های بین آنها حداقل m باشد، می‌توان از راه‌های زیر برای تسریع پیدا کردن جواب استفاده کرد:

۱. حذف گره‌های کم سن و سال با در نظر گرفتن حداقل سن برای کاندیداهای هر کشور.
۲. پیش پردازش گره‌های هر کشور و حذف گره‌های بدون یال یا یال‌های کمتر از c.
۳. استفاده از ترتیب جستجوی حریصانه: گره با بیشترین درجه را در هر کشور انتخاب کرده و تعداد یال‌های آن را می‌شماریم. در صورتی که تعداد یال‌ها بیشتر از m باشد که جواب مثبت است. اگر نه جستجو برای جواب را به ترتیب با جایگزین کردن گره با دومین بیشترین درجه در هر کشور ادامه داده تا، جواب مثبت پیدا شده، یا تمام حالات آزمایش شده باشد و در هیچ حالتی تعداد یال‌ها بیش از m نشود.

۳. [۲۸ نمره] ویروس کش: ایده این سوال از مصاحبه طراح سوال در مایکروسافت [12] است. مساله‌های زیر را به صورت رسمی درآورده [13] و برای حل آنها الگوریتم بهینه ارائه دهید.

(آ) هزار فایل مایکروسافت ورد متفاوت داریم که توسط ویروس `W97M.Marker.o` با یک الگوی ثابت ولی نا معلوم دستکاری شده‌اند (مجموعه A). علاوه بر اینها هزار فایل ورد متفاوت دیگر داریم که ویروسی نیستند (مجموعه B). فرض بر این است که تمام فایل‌های ویروسی دارای این الگوی ثابت هستند. ولی فایل‌های ورد به طور طبیعی به خاطر فرمت فایل، اشتراکاتی با هم دارند که ربطی به ویروس ندارد. هدف پیدا کردن بزرگترین زیر رشته مشترک بین مجموعه A است که در هیچکدام از فایل‌های مجموعه B نباشد. برای حل این مساله نام بردن یا اشاره کردن به اسم الگوریتم‌های موجود در اسلایدها کافیست (لازم است در حدی جزئیات را بگویید که مشخص شود دقیقا چه الگوریتمی مورد نظرتون هست). پیچیدگی محاسباتی و حافظه‌ای راه حل خود را بنویسید.

بیان رسمی مساله:
قسمت اول: پیدا کردن طولانی‌ترین زیر رشته مشترک بین k رشته با طول حداکثر L.
الگوریتم: می‌توان `Suffix Tree` [14] هر رشته با پیچیدگی محاسباتی [15] $O(L\log L)$ پیدا کرد. گره مشترک در میان تمام درخت‌ها با بیشترین عمق که تمام مقادیر یال‌ها بین ریشه تا آن گره بین تمام درخت‌ها مشترک باشند، بزرگترین زیر رشته مشترک است. با پیمایش

[11] `constant factor`
[12] `Microsoft Mesh Networking Group`
[13] Formalize

موازی تمام درخت‌ها با پیچیدگی محاسباتی $O(L \times k)$ می‌توان این گره را پیدا کرد. برای توضیحات بیشتر/متفاوت به این لینک مراجعه کنید. پیچیدگی حافظه‌ای برابر است با میزان حافظه لازم برای ساخت و نگهداری تمام درخت‌ها: $O(L \times k)$.
قسمت دوم: جستجوی رشته P در k رشته با حداکثر طول L.
الگوریتم: تمام رشته‌ها را در با کاراکترهای «جدید» و «یکتا» به هم متصل کرده تا رشته حرفی جدید به طول $L \times k + (k-1)$ بدست آید. سپس `Suffix Tree` معادل با پیچیدگی محاسباتی $O(kL \times \log kL)$ پیدا می‌کنیم. با استفاده از این درخت جستجوی P در مجموعه k رشته برابر با L خواهد بود.
تلفیق قسمت اول و دوم: در الگوریتم بخش اول، هنگام پیشمایش موازی درخت‌های پسینه ساخته شده برای مجموعه A برای پیدا کردن عمیق‌ترین گره، تنها زمانی گره را به عنوان جواب در نظر می‌گیریم که با توجه به الگوریتم بخش دوم، زیر رشته مورد نظر در مجموعه رشته B موجود نباشد. در نهایت پیچیدگی محاسباتی تلفیق قسمت اول و دوم برابر است با:

- پیچیدگی محاسباتی ساخت k درخت پسینه برای رشته‌های با طول حداکثر L برابر است با $O(k \times L \times \log L)$
- پیچیدگی محاسباتی ساخت یک درخت پسینه برای رشته‌ای با طول $L \times k$ برابر است با $O(kL \times \log kL)$
- پیچیدگی محاسباتی تلقیق الگوریتم قسمت اول و دوم برابر است با $O(L \times k \times L)$

در نتیجه پیچیدگی محاسباتی در مجموع برابر است با:

$$O(k \times L^2 + kL \times \log kL)$$

چنانچه از الگوریتم خطی برای ساخت درخت پسینه استفاده شود پیچیدگی محاسباتی نهایی برابر می‌شود با:

$$O(k \times L^2)$$

پیچیدگی حافظه‌ای نیز برابر است با کل حافظه لازم برای ذخیره تمام درخت‌های پسینه که برابر است با:

$$O(k \times L)$$

(ب) گروه تحقیقات شرکت الگوی یک میلیون ویروس را استخراج کرده است. هدف پیدا کردن تمام فایل‌های یک کامپیوتر است که دارای یکی از این الگوها می‌باشند. پیچیدگی محاسباتی و حافظه‌ای راه حل خود را بنویسید.

بیان رسمی: تعداد k الگوی با طول حداکثر P و n رشته به طول حداکثر L موجود می‌باشند. تمام رشته‌هایی که دارای یکی از k الگو می‌باشند را پیدا کنید.
الگوریتم: k الگو را در یک Trie ذخیره کرده، سپس به ازای هر رشته الگوها را توسط Trie در آن جستجو می‌کنیم.
پیچیدگی محاسباتی:

- ساخت Trie : $O(k \times P)$
- جستجوی الگوها در n رشته : $O(n \times L \times P)$

پیچیدگی محاسباتی نهایی:

$$O(k \times P + n \times L \times P)$$

پیچیدگی حافظه‌ای برابر است با میزان حافظه مورد نیاز برای نگهداری Trie و حداکثر یک رشته:

$$O(k \times P + L)$$