شهرزاد سبز* تکلیف دوازدهم نظریه زبان - استاد حجت ۸۱۰۱۰۰۲۷۲

الف اشتباه است - سوالی در رابطه با NP-complete است که NP-hard و NP را قطعاً نمی‌توانیم بگوییم که در مورد A است. به دلیل اینکه در این مسئله، A نسبت به B کاهش پیدا کرده است ولی توانایی نداریم که بگوییم در هر گونه سوال، NP نسبت به A کاهش پیدا خواهد کرد و حتی توانایی نداریم بگوییم که پاسخ A با order چند جمله‌ای حل می‌شود. یک مثال نقض برای نقض کردن این مورد توانایی داریم که A را یک زبان که (NP است) ولی NP-hard نیست، فرض کنیم به این دلیل که B ← NP-hard است. و همچنین $A \leq_p B$ از طرفی باید این را هم بدانیم که A NP-complete نیست.

ب اشتباه است - شبیه مثال قبلی ۲ حالت قطعی داریم. در صورتی که سوال در مورد NP باشد قطعاً آن سوال درست، برقرار نیست - حالت دوم در صورتی که NP-hard باشد، سوال پایدار و پابرجا و برقرار است. مثال نقض: یک سوالی که NP-hard است ولی NP نیست را فرض می‌کنیم و در این حالت $A \leq_p B$ پایدار و برقرار است ولی می‌دانیم که B NP-complete نیست

ج صحیح - چون در مواردی که NP-complete است، ما می‌دانیم که NP است، هر گونه NP را می‌توانیم به یک NP-complete در زمان چند جمله‌ای، کاهش دهیم

د صحیح است - در نظر می‌گیریم که B یک زبان NP است، به این دلیل که NP = P بنابراین B با زمان چند جمله‌ای Deterministic Decider دارد - در نظر می‌گیریم که A در عوض ورودی w_a accept می‌کند، در عوض [در برابر] ورودی w_r reject می‌کند. اکنون در عوض ورودی w به B، در مرحله‌ی اول $n(w)$ را انجام می‌دهیم. در صورتی که مورد قبول واقع شد، w_a را به A واگذار می‌کنیم که آن را n نام‌گذاری کردم [به دلخواه]

و در صورتی که مورد قبول واقع نشد، w_r را به A واگذار می‌کنیم. $f(w) \in L_A$ آن وقت $w \in L_B$

و در نهایت: $B \leq_p A$

$w \in L_B$ آن وقت $f(w) \in L_A$

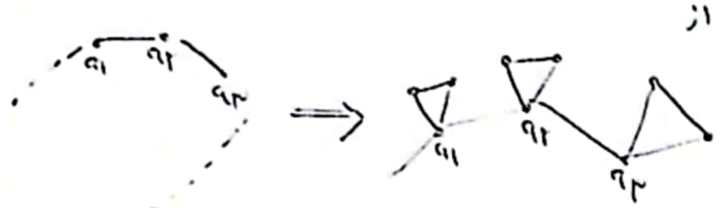
۲ ثابت کنید که $mode\ p$ در دسته مسائل P قرار دارد؟ در نظر می‌گیریم که b یک عدد n_1 بیتی باشد و b را به حالت گسترده می‌نویسیم و می‌پنداریم که اعداد b_1 و b_2 و b_3 ... b_n ایجاد شوند. اعدادی ایجاد می‌شوند که بیشتر آنها برابر برابر ۲ اند.

$$n_b \cdot a^b = a^{b_1+b_2+b_3+\dots+b_n} = a^{b_1} \cdot a^{b_2} \cdot a^{b_3} \dots a^{b_n}$$

a^{b_i} را از روی ۲/۱۵/۰۰۰۲ a را بر روی ماشین تورینگ Laghi n_a^2 طول می‌کشد که این یک عدد چند جمله‌ای است n عدد داریم که لازم است باقی آنها را حساب کنیم و می‌دانیم که Polynomial O ممکن است، بنابراین mod p توسط polynomial O پاسخ می‌دهیم و قطعاً در دسته‌ی P قرار دارد.

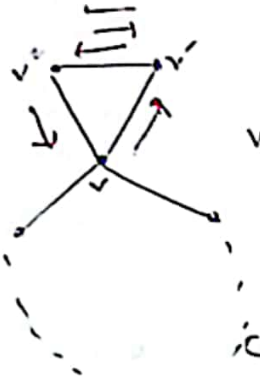
۳ در عوض یک راس دلبخواهی، v در اطراف اصلی دو راس v' و v'' را اضافه می‌کنیم و بین v و v' و v'' یال‌ها را ... ساختیم

۳ اکنون درصورتی که گراف حاصل گذر ویل داشته باشد، از هر گوشه راس v گراف اصلی دو بار می‌گذریم به این دلیل که v راس به شاخه است که کل گذر را به ۲ عدد گذر بسته‌ی جدا جدا تقسیم می‌کند



که باقی‌مانده‌ی گراف

اکنون درصورتی که در عوض همه‌ی راس‌ها در گراف اصلی شمای دورشان با v' و v'' را حذف می‌کنیم و از بین هر یک دور در گراف اصلی دست پیدا می‌کنیم و می‌رسیم که در عوض هر راس به جایی که [illegible] گذرهایش وجود دارد را حذف می‌کنیم. از هر راس یک بار گذر می‌کنیم و دور همیلتونی داریم. درصورتی که گراف اصلی دور همیلتونی داشته، در گراف حاصل گذر ویل داریم به این دلیل که به هر راسی که گذر می‌کنیم یک دور به شکل زیر با v' ما می‌زنیم و سپس به راس بعدی به دور همیلتونی می‌رسیم؛



بنابراین گراف ورودی دور همیلتونی دارد و درصورتی که $v \to v' \to v'' \to v' \to v$
اگر و تنها اگر گراف حاصل گذر ویل داشته باشد و تغییر گراف ورودی به گراف حاصل از order مقدار راس‌های گراف است. به این دلیل که در عوض هر راس، ۲ راس دیگر و ۳ یال دیگر را اضافه خواهیم کرد. بنابراین دور همیلتونی، به گذر ویل کاهش پیدا خواهد کرد بنابراین گذر ویل np-hard است.

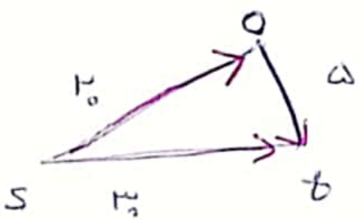
۴ درنظر می‌گیریم که مجموعه‌ی U به عنوان لیست اعداد و مجموعه‌ی S به عنوان مجموعه‌ای از زیرمجموعه‌های S، و علاوه بر آن، عدد t به سوال set cover داده شده است. مجموعه‌ی U را به عنوان لیست پهلوانان و مجموعه‌ی S را لیست پادشاه‌هایی فرض می‌کنیم که هر پادشاه، لیستی از پهلوانان که عضو شده را نشان می‌دهد. مقصود این است که t را زیرمجموعه حفظ کنیم و لازم است hs-t را به پهلوانان که به عنوان ورودی دهیم. این تغییر از اردر ۱ طول می‌کشد چون تنها تغییری که می‌کند، تبدیل t به hs-t است.
پس نتیجه می‌گیریم تبدیل polynomial است. با ورودی دادن تبدیل بالا به سوال پهلوانان، درصورتی که خروجی اکسپت کنیم، به این معنی است که می‌توانیم با حفظ کردن و نگه داشتن t، پادشاه را قانون و شرط مسئله را رعایت کنیم. بنابراین set cover هم اکسپت می‌شود و اگر خروجی reject [ریجکت] شد، به این معنی است که نتوانستیم و خروجی set cover هم ریجکت می‌شود. پس سوال پهلوانان np-hard است. نکته: هر جواب احتمالی را توانایی داریم در زمان polynomial [درستی] هم بررسی کنیم؛ و این سوال np-complete هم می‌باشد.

۵ سوال: ۲ مسئله یافتن کوتاه‌ترین مسیر در گراف را درنظر بگیرید. [الف] یک عدد راس t' و یال $t'-t$ با وزن $+\infty$ و s بزرگ‌تر از تمام یال‌های دیگر، به گراف را اضافه کردیم. و گراف حاصل را به ماشین A سوال B خواهیم داد. هر مسیری که از s به t' [به جای s و t، s و t' را به ماشین واگذار می‌کنیم] از $t' \longleftarrow t$ خواهیم گذشت و به این دلیل که وزن این یال max است [بیشینه]، اندازه‌ی همین

یال را حذف می‌کنیم و از بین می‌بریم، خروجی پاسخ t' و A را حذف می‌کنیم و از بین می‌بریم و در نهایت پاسخ دقیقا شبیه پاسخ ماشین A شد.

ب) یک گراف معادل گراف ورودی به وجود می‌آوریم. در صورتی که یک راس در گراف اصلی v بنامیم، در گراف copy شده راس متناظر اش را v' اسم گذاری می‌کنیم. و در حال حاضر در صورتی که در گراف اصلی راس u به v یال داشته باشد از u به v' یالی با وزن صفر خواهیم کشید. و اینگونه می‌شود که در میان گراف اصلی و گراف copy شده یال‌هایی با وزن صفر داریم که جهت همه آنها از گراف اصلی به گراف کپی شده است. بنابراین در صورتی که این مجموعه را به حل کننده A واگذار کنیم، توانایی دارند همه‌ی مسیرها و راه‌هایی را که وزن یکی از یال‌ها صفر فرض می‌شود، هم چک کند. بنابراین می‌تواند مسیر را همانی که B خروجی می‌دهد را می‌یابد و پیدا می‌کند ← check

این کاهش خطی می‌باشد چون گراف را copy می‌کنیم و میان ۲ گراف e یال خواهیم کشید که مقدار و تعداد آنها مساوی مقدار یال‌های گراف اصلی است. پس بنابراین order اُردر تبدیل در order مقدار یال‌ها می‌باشد.



t' ـش s
t' ـش ۵
۵ ـش s

به این دلیل که reduction ها، polynomial با Algorithm A انجام داده است.
← چند جمله

بنابراین $B \leq_p A$

$\boxed{4}$ برای حل سوال یک تعداد مسئله 3-sat را به مسئله صورت سوال کاهش می‌دهیم. به این معنی است که
که sar tree
نشان می‌دهیم
ثابت می‌کنیم مجموعه این مسئله در دسته‌ی np-complete قرار دارد. هر یک از راس‌ها مجموعه‌ی iام
را درون آن می‌گذاریم. بنابراین در هر مجموعه حداقل باید یک یک وجود داشته باشیم. رنگ سبزها اندیس ۱ (لیبل یک) و رنگ
قرمزها اندیس ۰ (لیبل ۰) دارند.

6. set spliting

تقسیم مجموعه ای در NP به دلیل رنگ آمیزی عناصر میتوانند شناخته بشن که ویژگی های خواسته شده در polynomial time را دارند.

set spliting به 3-sat reduction polynomial می دهیم. به صورت روبه رو: ما یک کاهش polynomial time به نام P داریم که از 3-sat به set spliting است، و آن را به صورت زیر شرح می دهیم:

یک 3cnf داده شده که $S = \{x_1, \bar{x}_1, \ldots, x_m, \bar{x}_m, y\}$ تعریف می کنیم و به عبارت دیگر S شامل یک element میشه [۲ تا برای هر متغیر که برای هر حرف] و یک element به خصوص y. برای هر جمله ی c_i داخل ϕ اجازه بدید که $subset\ c_i$ S باشند در حالیکه شامل element هایی است که match هستند با yes. همچنین ما subset های دیگر C_{x_i} را برای هر حرفی که شامل element x_i و $\bar{x}_i$ اضافه می کنیم. سپس $C = \{c_1, \ldots, c_l, c_{x_1}, \ldots, c_{x_m}\}$

اگه تخصیص مورد قبول واقع بشه، یک علامت مورد قبول را در نظر میگیریم. اگه ما همه ی حروف درست را قرمز کنیم و همه ی حروف غلط را آبی کنیم، و y را آبی کنیم سپس هر زیر مجموعه ی c_i از S حداقل یک عنصر قرمز خواهد داشت. (بخاطر اینکه از تخصیص به واسطه برخی حروف روشن شده) و همچنین B شامل یک عنصر y آبی میشه $A \le B$ [اگه A را به B تا مسئله بدیم]. و همچنین هر subset C_{x_i} دقیقا یک عنصر قرمز رنگ و یک عنصر آبی رنگ دارد. بنابراین این سیستم $\langle S, C \rangle \in set\_spliting$ هست.

اگه $\langle S, C \rangle \in set\_splitting$ باشد، سپس به رنگ آمیزی S نگاه می کنیم. اگر که ما آن حروف را درست باشند یک رنگ متفاوتی از y دارند. و آنهایی که false (نادرست) باشند، آن یک رنگ مشابه y را دارد. پس نتیجه میگیریم که یک علامت مورد قبول برای تخصیص به دست می آوریم.

<u>در واقع</u>: میخواهیم مسئله ی 3sat را به این مسئله کاهش دهیم. فرض می کنیم 3cnf به ما داده شده. S را می سازیم به گونه ای که $S = \{x_1, \bar{x}_1, \ldots, x_m, \bar{x}_m, y\}$ در عوض هر clause ای که در 3cnf داریم، یک مجموعه ای ۴ تایی تشکیل می دهیم که ۳ عضو آن clause را به همراه y داشته باشد. [y جز 3cnf داده شده نیست] اکنون این مجموعه های ساخته شده را در X قرار می دهیم. به X مجموعه های $[x_1, \bar{x}_1]$، $[x_2, \bar{x}_2]$، ...، $\{x_m, \bar{x}_m\}$ را اضافه می کنیم. مجموعه ی S و مجموعه ی X را به عنوان ورودی به این مسئله می دهیم. می توان گفت که اگه جواب برای 3sat پیدا شود، آن literal هایی که درست هستند را قرمز کنیم و آنهایی را که false هستند آبی کنیم و y را هم آبی کنیم، جواب این مسئله هم می باشد. چراکه $\{x_i, \bar{x}_i\}$ ها که یک آبی و یک قرمز دارند و از آنجایی که مسئله 3sat حل شده، در حداقل یک در هر قسمت true هست یعنی حداقل یک قرمز در هر مجموعه و از آنجایی که y در همه ی مجموعه ها هست و آبی هست، یعنی در همه ی مجموعه ها حداقل یک قرمز و آبی را داریم. پس این مسئله NP hard است. از آنجایی که چک کردن هر جواب در زمان چند جمله ای هست (چون باید m مجموعه را که مجموع n عنصر دارند را چک کنیم) پس مسئله NP-complete است.

۷) مسئله A که مسئله اول است که همان مسئله گروه کردن مجموعه‌هاست قابل کاهش است به مسئله‌ای که بتوانیم مساحتی را $\{1 \times (4 \cdot n) \mid n \in S\}$
با مستطیل را با مستطیل‌هایی به فرم ... فرض کنیم یا در واقع بتوانیم این مستطیل‌ها را به فرم ... به هم بچسبانیم و بسته
بندی کنیم یا در واقع بلوک‌هایی به این ترتیب ایجاد کنیم. $2 \times (\sum_{n \in S} 2^{n})$
در این حالت می‌توانیم پکیج‌های ایجاد شده را از بالا به یک مجموعه و از پایین مجموعه دیگر مرتبط کنیم و مسئله را اینگونه
حل کنیم. پس این ۲ مسئله به هم قابل تبدیل شدن هستند و می‌توان مسئله گروه بندی را به دیگری کاهش داد پس کاهش انجام
شده چون از A به عنوان یک مسئله NP-complete هست پس مسئله B یک مسئله NP hard
است

در واقع ..: فرض می‌کنیم مجموعه T ورودی مسئله A است، برای این reduction ما به ازای هر عدد موجود
در مجموعه T، [ابتدا n را می‌گذاریم] کاشی‌ای به عرض ۱ و طول $2n$ می‌سازیم. فرض می‌کنیم که مجموع اعداد
مجموعه T برابر x می‌باشد. اکنون مجموعه کاشی‌های ساخته شده را به همراه اتاقی به طول $\frac{2n}{2}$ و عرض
۲ به مسئله B می‌دهیم، می‌توانیم بگوییم که در صورتی که مسئله A، ورودی T را بپذیرد، مسئله B هم ورودی
ساخته شده را می‌پذیرد. چون کوچکترین طول کاشی موجود می‌تواند ۲ باشد پس آن را به صورت عرضی نمی‌توانیم
قرار بدهیم. در صورتی که بتوانیم کاشی‌ها را بچینیم، یعنی ۲ ستون به عرض یک کاشی پر شده است که متشابه با
برای تقسیم مجموعه به ۲ گروه با جمع برابر است. پس در صورتی که [متناظر] $A \leq B$ را مطرح داریم، از آنجایی که NP-C A
است B هم NP-hard است. تبدیل هم تبدیلی به صورت خطی است به این دلیل که اگر n عدد در مجموعه
T داشته باشیم، فقط باید یک مجموعه n تایی کاشی تولید کنیم که $O(n)$ می‌باشد.