علیرضا کریمی 810101492 1 روز توکن

سوال ۱

الف

1) خیر؛ چون ممکن است جزو مسائل NPC نباشد. چون تمام مسائل P و NP را می‌توان به NPC کاهش داد و در صورتی که تنها یک NPC را بتوان به صورت چندجمله‌ای حل کرد، تمام NPها را می‌توان به آن کاهش داد و آن‌ها را نیز به صورت چندجمله‌ای حل کرد و NP و P آنگاه با هم برابر می‌شوند. اما این NPC نیست (یعنی کاهش با اردر چندجمله‌ای)

ب

2) خیر؛ چون مسائل NP تحت عملیات star بسته هستند و باز NP می‌ماند

(برای اثبات بسته بودن کافی است که ماشین تورینگ فعلی ما T نام دارد و می‌خواهیم از روی آن T' را بسازیم به گونه‌ای که اگر T، A را بپذیرد، T' باید A^* را بپذیرد. پس T' را اینگونه تعریف می‌کنیم که ε = w ورودی را بپذیرد. در غیر اینصورت باید به صورت non deterministic تمام تقسیم‌بندی‌های مختلف را روی w اجرا کن و چک کن که هر یک از قطعه مورد قبول ماشین تورینگ T است یا نه. اگر همه توسط T پذیرفته شدند، T' هم می‌پذیرد در غیر اینصورت در دیگر از حالات نمی‌پذیرد)

در واقع اگر باز تحت عمل star باشد، تضمین نمی‌شود که در اردر چندجمله‌ای حل شوند.

پ

3) بله؛ CFGها جزو P هستند (طبق تعریف در کتاب سیپسر). از آنجا که 3SAT را توانستیم با اردر P حل کنیم و همه مسائل NP و P هم به این مسئله (که NPC است) می‌توانند با اردر چندجمله‌ای کاهش پیدا کنند، پس مسائل NP هم با اردر چندجمله‌ای حل خواهند شد و NP = P می‌شود

ت

4) خیر؛ چون دوباره همون NP می‌شود و دوباره non deterministic polynomial می‌شود و دوباره باید با اردر exponential تبدیل شود

ث

5) بله؛ NFA قابل تبدیل به DFA است. DFAها هم با اردر چندجمله‌ای حل می‌شوند. از طرفی NFAها زیرمجموعه CFGها می‌شوند و می‌دانیم CFGها هم با اردر چندجمله‌ای حل می‌شود. پس دوباره مثل قسمت پ، برای مسئله NPC 3SAT، یک راه حل چندجمله‌ای داریم و همه مسئله‌های NP هم به آن قابل کاهش است (با اردر چندجمله‌ای) و پس NPها هم چندجمله‌ای می‌شوند و NP=P

(3SAT یا برای راحتی کار SAT)

سوال 2)

برای اثبات NP hard بودن آن، مسئله 3SAT را که NP-hard است به آن کاهش می‌دهیم. در یک کاهش با زمان چندجمله‌ای. اگر فرض خلف بگیریم که ماشین M، از زمانی NP نیست و به صورت چندجمله‌ای حل می‌شود، پس با کاهش 3SAT به آن، 3SAT هم باید به صورت چندجمله‌ای حل شود که این موضوع هنوز اثبات نشده. از طرفی چون 3SAT خودش NP-hard است، اگر NP-hard را به مسئله‌ای دیگر کاهش دهیم آن نیز NP hard خواهد شد. پس تنها باید کاهش را به گونه‌ای انجام دهیم تا ورودی 3SAT به ورودی ماشین M تبدیل شود. این کار را در ۳ مرحله باید انجام دهیم.

1) ابتدا تمام B_j ها را به صفر initialize می‌کنیم ($1 \le j \le m$)

2) حال هر clause را مربوط به یک M_j می‌گیریم. مثلا داریم: ($x_1 \vee \bar{x}_2 \vee x_3$)
برای هر x_i ($1 \le i \le n$) یک a_{ij} مربوط به آن در آن کلاس در نظر بگیرید. درصورتی که x_i داشتیم، $a_{ij}=1$ و درصورتی که $\bar{x}_i$ داشتیم $a_{ij}=-1$ را قرار می‌دهیم و درصورتی هم که x_i اصلا در آن clause نبود (فرض کنید داریم SAT می‌رویم و بیشتر از 3 تا داریم. درواقع بهتر است بگوییم n-SAT)، آن گاه آن را $a_{ij}=0$ قرار می‌دهیم

3) درحالتی که $\bar{x}_i$ داشتیم و $a_{ij}=-1$، آنگاه از b_j یک واحد کم می‌کنیم.

یعنی مثلا اگر داشته باشیم: $(x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee x_3 \vee x_4)$ عبارت بولین ورودی

$\longrightarrow$ $a_{11}=1,\ a_{21}=-1,\ a_{31}=1,\ a_{41}=0$ $\quad$ $a_{12}=1,\ a_{22}=1,\ a_{32}=1,\ a_{42}=0$

$$\longrightarrow (x_1 - x_2 + x_3 > \underset{B_1=-1}{-1}) \wedge (x_1 + x_3 + x_4 > \underset{B_2=0}{0})$$

اگر درستی این عبارت برقرار باشد، معادل این است که درستی عبارت بولین برقرار است، و اگر درستی این عبارت، چندجمله‌ای انجام شود (متعاقبا این است که P است) پس SAT هم P می‌شود چون به آن کاهش یافته که تناقض

پس M، NP-hard است (و چون که SAT که NP hard است به آن کاهش دادیم پس M، NP hard می‌شود طبق تعریف در جزوه)

سوال

3) مثل سوال 2، برای اینکه ثابت کنیم np-hard است، باید یک مسئله np-hard را به آن کاهش دهیم. در اینجا از مسئله دور همیلتونی (HAMPATH) استفاده می‌کنیم، یعنی می‌خواهیم بدانیم در گراف G، یک مسیر همیلتونی از s به t وجود دارد یا نه و این کار را با تغییر ورودی به حالتی که ورودی مسئله است یعنی یک گراف با راس‌های شکلات‌دار، انجام می‌دهیم. از آنجایی که این تغییر ورودی باید در زمان چندجمله‌ای باشد، درصورتی که مسئله گراف شکلات‌دار را در چندجمله‌ای حل کنیم، عملا انگار HAMPATH را در چندجمله‌ای حل کردیم که این هنوز اثبات نشده است که HAMPATH جزو مسائل P است (فرض کردیم np است). پس مسئله گراف شکلات‌دار باید خودش np-hard باشد.

برای تغییر ورودی، بیاییم روی تمام راس‌های گراف G، یک شکلات قرار می‌دهیم به جز راس s که 2 تا قرار می‌دهیم. از طرفی از راس t به راس جدید یال می‌زنیم (مثلا t') و روی آن هم یک شکلات می‌گذاریم (در آخر می‌گویم چرا) دلیل آن این است که مطمئن شویم در مرحله آخر بر روی راس t می‌رسیم و در راس t خاتمه می‌یابد. از راس t'

حال فرض کنید این گراف تغییر یافته شکلات‌دار را به عنوان ورودی به ماشین M صورت سوال بدیم و فرض کنید که تمام شکلات‌ها به جز یک شکلات خورده شود. خب یعنی چه؟ یعنی اینکه از s شروع کردیم (چون تنها راس 2 شکلات‌دار در ابتدا همان s است) بعد 2 شکلات آن را برداشتیم و یکی را به همسایه آن دادیم. در مرحله بعد تنها می‌توانیم به سراغ آن همسایه‌ای برویم که با دریافت 1 شکلات الان مجموعا دو شکلات دارد. s هم الان تعداد شکلات‌هایش صفر شده. همینطور راس‌ها را پشت سر هم می‌گذاریم و در هر مرحله تنها یک راس وجود دارد که 2 شکلات دارد و تنها می‌توان آن را پیمود. از آنجا که تمام راس‌ها 1 شکلات داشته‌اند، همه آن‌ها را باید به همین روش بپیمود (یعنی برای هر راس یک شکلات از خودش و یک شکلات از همسایه‌اش گرفته) در مرحله آخر به t' می‌رویم که باید دوتا شکلات داشته باشد و آن دوتا را مصرف می‌کنیم و یکی به t می‌دهیم (چون تنها همسایه t' بود) و الان تنها یک راس t با یک شکلات داریم و بقیه صفرند. هر راس را هم اینگونه دقیقا طی کردیم

(t' را برای این گذاشتیم که یعنی در مرحله آخر سراغ آن هم t رفتیم و از روی آن قرار نگیرد تنها یک شکلات آن هم روی t می‌نهیم که در مرحله آخر آنجا بودیم) و دور همیلتونی وجود دارد.

ادامه 3) سپس HAMPATH را به M کاهش دادیم اما همانطور که در ابتدا گفتیم، M نمی تواند از مسائل P باشد چون HAMPATH، P می شود پس فرض خلف باطل می شود و M از مسائل NP hard است چون HAMPATH که NP-hard است را به آن کاهش دادیم. به طور شهودی هم می توان فهمید که M، NP_hard است، از آنجا که وقتی دو شکلات موجود در S را می خورند، به صورت non deterministic تقسیم می گیرد یک شکلات را به چه کسی بدهد و برای هر حالت هم در اردر چند جمله ای step برای رسیدن به مقصد داریم.

یکی از

سوال 4) شرط کافی است که به عنوان مثال، اگر A ما boolean expression باشد، چک می کنیم که آیا هم زمان هم A و هم $\neg A$، satisfiable هستند یا خیر. درصورتی که بودند، یعنی اینکه هم حالتی وجود داشته که A درست می شده و هم حالتی وجود داشته که A غلط می شده و بنابراین همیشه درست یا همیشه غلط نیست. در غیر اینصورت هم می تواند همیشه درست یا همیشه غلط باشد.

سوال 5) باید A را به B کاهش دهیم تا ثابت شود B نیز NP hard است. یعنی باید ورودی ماشین A را با یک تغییرات در اردر چند جمله ای تبدیل کنیم به ورودی ماشین B. برای این کار، می دانیم که مجموع اعضای زیر مجموعه ی انتخابی از t بیشتر نیست. یعنی تک تک اعضای آن از t کوچک مساوی هستند.

حال به ازای هر عدد x_i که از t کوچک مساوی است و در S قرار دارد، می آییم دو عضو x_i و $2t-x_i$ را به یک مجموعه جدید با نام S' اضافه می کنیم (دلیلش رو جدا می گم) (اگر $x_i=t \leftarrow 2t-x_i = 2t-t=t$

یک بار اضافه می شود به S')

پس الان در مجموعه S' خواهیم داشت:

$$S' = \{x_1, x_2, \dots, x_n, 2t-x_1, 2t-x_2, 2t-x_3, \dots, 2t-x_n\}$$

$$1 \leq i \leq n \; ; \; x_i \leq t \qquad x_i \leq t \rightarrow 2t - x_i \geq t \quad 1 \leq i \leq n$$

حال اگر S' را به ماشین B بدهیم چه اتفاق خواهد افتاد. از طرفی داریم:

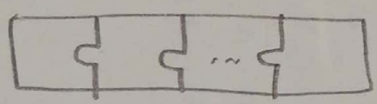
$$\sum_{i\in S'} = \frac{1}{|S'|}\sum_{i\in S'} i = \frac{1}{2n}\left(x_1+x_2+\cdots+x_n+(2t-x_1)+(2t-x_2)+\cdots+(2t-x_n)\right)$$

$$= \frac{1}{2n}\, n\times 2t = t$$

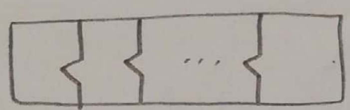
یعنی کافی است زیرمجموعه‌ای مانند S'' از میان اعضای S' پیدا کنیم که جمع آنها t باشد. این اعضا قطعاً از میان اعضای x_i $(1\le i\le n)$ انتخاب می‌شوند چون این اعضا کوچک‌تر مساوی t هستند و اعضای $2t-x_i$ $(1\le i\le n)$ بزرگ‌تر مساوی t هستند که به کار ما نمی‌آید. از طرفی تمام اعضای S'' در S نیز هستند (چون تمام x_iها در ابتدا در S بودند). پس اگر بتوان S'' را پیدا کرد معادل آن زیرمجموعه‌ای از S وجود داشته که مجموع آنها t بوده، و به این شکل A قابل حل به کمک B بوده. اگر B چندجمله‌ای حل شود پس A نیز چندجمله‌ای حل می‌شود (غربال اینکه هر عضو S از t کمتر مساوی است یا نه در اردر چندجمله‌ای انجام می‌شود. پس تغییر ورودی A به ورودی B از اردر چندجمله‌ای است) اما A، NPC است $\Longrightarrow$ B نیز NPC خواهد بود

6)

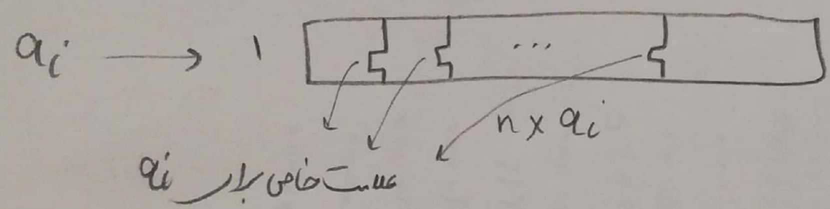
الف) اگر قرار باشد که $n=4m$ عدد را به m گروه تقسیم کنیم که هر کدام ~~۴ عضو داشته باشند~~ مجموع هر کدام از آن زیرمجموعه‌ها یکسان باشد و برابر $t=\frac{\sum}{m}=\frac{\sum}{n/4}$ باشد، به ازای هر عدد a_i، a_i، $n\times a_i$ تا کاشی unique درست می‌کنیم که مربوط به آن a_i خاص باشد. مثلا برای یک a_i داشته باشیم



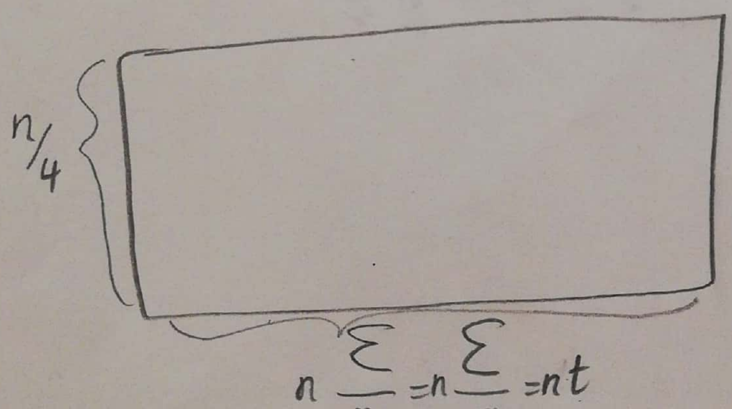
و برای یک a_i دیگر کاشی‌ها با این شکل باشد:



این تضمین می‌کند که کاشی‌های مربوط به a_i، فقط با یکدیگر match می‌شوند و با دیگر کاشی‌ها match نمی‌شوند. از طرفی به ازای هر a_i، به تعداد na_i از این کاشی‌ها درست می‌کنیم. (دلیلش رو بعدا می‌گم)



حال مستطیل حاصل را این گونه می‌سازیم



برای قرار دادن کاشی‌ها به طول $n\times a_i$، چون $na_i > n/4$ است، آنها را نمی‌توان عمودی چید و باید افقی چید. پس می‌توان دید و گفت که $n/4$ ردیف داریم که طول هر کدام از آنها برابر nt است که در واقع n برابر همان مجموعه اعضایی است که قرار است مجموع آن‌ها t باشد (برای همین طولش nt است) از آنجا هم که a_i از اردر چندجمله‌ای است، تعداد na_i کاشی به ازای هر کدام هم از اردر چندجمله‌ای ساخته می‌شود

۶ ب) خیر؛ چون خود تبدیل هم باید از اردر چندجمله‌ای باشد و اگر نه در اردر چندجمله‌ای نباشد

و مثلا در اردر نمایی باشد، کاهش‌های تولیدی هم از اردر نمایی است و برای تبدیل مسئله A به B

باید به اندازه نمایی هزینه کنیم.