تمرین ۶ – پاسخ

مسئله اول

الف) کلاس پیچیدگی NP شامل تمام مسائل تصمیم‌گیری است که توسط ماشین تورینگ غیرقطعی در زمان چندجمله‌ای قابل حل هستند. به زبانی دیگر، L متعلق به کلاس NP است اگر و تنها اگر الگوریتم A در زمان چندجمله‌ای وجود داشته باشد به گونه‌ای که با داشتن y به عنوان certificate بتوان درستی آن را با آن الگوریتم بررسی کرد

ب) مسئله A قابل کاهش به مسئله B است اگر الگوریتمی برای حل مسئله B در زمان چندجمله‌ای وجود داشته باشد که بتوان از آن در حل مسئله A در زمان چندجمله‌ای استفاده کرد. و می‌نویسیم

$A \leq_p B$

ج) NP Hard: مسئله H عضو کلاس np-hard است اگر و تنها اگر تمام مسائل L عضو np قابل کاهش به H در زمان چندجمله‌ای باشند.

د) NP Complete: مسئله C عضو کلاس np-complete است اگر و تنها اگر عضو np باشد و تمام مسائل دیگر عضو np قابل کاهش به این مسئله در زمان چندجمله‌ای باشند.

مسئله دوم

الف) باید نشان دهیم که این مسئله عضو کلاس NP است:

فرض کنیم مجموعه s’ داده شده و ادعا می‌شود جواب مسئله است --> کافی‌ست اعضای s’ را با هم جمع کنیم و چک کنیم آیا حاصل صفر می‌شود یا خیر --> در زمان چندجمله‌ای قابل انجام است

ب) مسئله ما انتخاب زیرمجموعه‌ای از S با مجموع t است <S,t>

حال این مسئله را به SAT-3 که قبلا نشان داده بودیم np-complete است کاهش می‌دهیم:

فرض کنید f فرمولی با n عبارت و m متغیر است. به ازای هر متغیر، برداری به طول m+n در نظر می‌گیریم و به صورت زیر آن را پر می‌کنیم:

M خانه اول برای متغیرها و از خانه m+1 تا m+n برای عبارت‌ها: [x1,x2,x3,x4;c1,c2,c3]

$f: (x2 \vee x3 \vee \sim x4) \wedge (x1 \vee \sim x3 \vee x4) \wedge (x1 \vee \sim x2 \vee x4)$

x1: [1,0,0,0;0,1,1] ~x1: [1,0,0,0;0,0,0]
x2: [0,1,0,0;1,0,0] ~x2: [0,1,0,0;0,0,1]
x3: [0,0,1,0;1,0,0] ~x3: [0,0,1,0;0,1,0]
x4: [0,0,0,1;0,1,1] ~x4: [0,0,0,1;1,0,0]

برای تعیین t، اگر تمام عناصر بردار t را ۱ درنظر بگیریم انگار دقیقا فقط یکی از هر متغیر انتخاب شده است که درست نیست. پس t را به صورت [3,3,3;1,1,1,1] در نظر می‌گیریم. ۱ به ازای تمام متغیرها و ۳ به ازای تمام عبارت‌ها. یعنی حداقل هر متغیر باید در یکی از عبارت‌ها باشد (xi, xi~)

پس اگر زیرمجموعه‌ای از بردارها را انتخاب کنیم که مجموعشان برابر با t شود، یعنی توانسته‌ایم f را satisfy کنیم.

مسئله سوم

الف) آیا از بین n شیء با وزن‌های wi و ارزش‌های ci زیرمجموعه‌ای یافت می‌شود که مجموع وزن آن‌ها حداکثر برابر B و مجموع ارزش آن‌ها بیشینه باشد؟

ب) ابتدا باید نشان دهیم این مسئله عضو NP است: زیرمجموعه s از اشیاء به عنوان جواب ارائه می‌شود. می‌توان مجموع وزن و ارزش آنها را در زمان چندجمله‌ای بدست آورد و چک کرد که آیا حداکثر وزن B و ارزش C را دارند یا خیر.

مسئله را به subset sum کاهش می‌دهیم: مجموعه w1 تا wn را داریم. زیرمجموعه‌ای از این n عضو را می‌خواهیم که مجموعشان برابر B شود.

چون subset sum عضو کلاس np-complete است، پس مسئله ما نیز np-complete است.

مسئله چهارم

الف) مسئله‌ی مطرح شده مسئله‌ی بهینه‌سازی است که در کلاس NP قرار نمی‌گیرد.

ب) عدد k داده شده است. آیا می‌توان مسیری به طول حداکثر k یافت به طوری که شرایط مسئله را براورده کند؟

ج) فرض کنیم مسئله‌ی بهینه‌سازی داده شده است. بر روی پارامتر k قسمت قبل binary search می‌زنیم. کوچکترین k را می‌یابیم که شرایط در مسئله‌ی decision صدق کند. جواب این مسئله جواب مسئله‌ی optimization است. الگوریتم باینری سرچ نیز از O(log p) است که p طول مسیر پاسخ است.

مسئله پنجم

مسئله را به independent set کاهش می‌دهیم: فرض کنیم پاسخی برای independent set با اندازه k داریم. مجموعه رئوس independent set را در نظر می‌گیریم. می‌دانیم در این مجموعه مسیری به طول ۱ وجود ندارد. حال مسیرهای به طول ۲ را درنظر می‌گیریم (با یک BFS به آن‌ها می‌رسیم). یک راس میان یکی از یال‌های مسیر اضافه می‌کنیم تا طول مسیر یک واحد بیشتر شود.

مسئله ششم

مسئله را به subset sum کاهش می‌دهیم. به این صورت که هر مثال ورودی جمع زیرمجموعه‌ای را به یک مثال چیدن بلوک‌ها تبدیل کنیم.

یک مثال جمع زیرمجموعه‌ای را اگر در نظر بگیریم، کافی‌ست به جای هر عدد k یک بلوک 1xk قرار دهیم. اگر جمع مورد نظر در مسئله جمع زیرمجموعه‌ای S باشد و جمع کل اعداد هم M باشد، اندازه جدول مسئله بلوک را 2xS می‌گیریم (با فرض S >= M/2. اگر S < M/2 باشد، اندازه جدول را 2xL می‌گیریم که L=M-S است). نهایتا یک بلوک دیگر 1xH باید اضافه کرد که مقدار H برابر است با

2S-M (با همان فرض S >= M/2)