آنالیز الگوریتم‌ها (۲۲۸۹۱)
[بهار ۹۹]

تمرین سری ۱۰ — موعد: سه‌شنبه ۱۵ اردیبهشت ساعت ۱۲

- سؤالات خود پیرامون تمرین را با alirtofighim@gmail.com مطرح کنید.

۱. با فرض $\mathbf{P} = \mathbf{NP}$ الگوریتمی چند جمله‌ای ارائه دهید که یک 3SAT به عنوان ورودی گرفته و در صورتی که ورودی ارضاپذیر باشد، مقداردهی متغیرها که 3SAT را ارضا کند خروجی دهد.

راهنمایی برای پاسخ: فرض کنید الگوریتمی با زمان اجرای $f(n)$ داریم که یک 3SAT با اندازه‌ی n گرفته و در زمان $f(n)$ می‌گوید که ارضاپذیر است یا خیر. همچنین چون مسئله‌ی 3SAT در کلاس $\mathbf{NP}$ است، فرض $\mathbf{P} = \mathbf{NP}$ نتیجه می‌دهد که الگوریتمی با زمان چندجمله‌ای نیز داریم و می‌توانیم فرض کنیم $f(n)$ نسبت به n چند جمله‌ای است.
حال یک 3SAT دلخواه به اندازه‌ی n شامل متغیرهای $x_1, \dots, x_k$ در نظربگیرید، الگوریتمی برای مقداردهی $x_1, \dots, x_k$ ارائه می‌دهیم.
S_0 را برابر با ورودی قرار دهید، S_0 را به الگوریتم بررسی ارضاپذیری 3SAT می‌دهیم، اگر جواب خیر داد چاپ می‌کنیم که 3SAT ارضا پذیر نیست و الگوریتم را تمام می‌کنیم، در غیر این‌صورت یک مقداردهی وجود دارد که S_0 ارضاپذیر است، حال k بار عملیات زیر را انجام می‌دهیم:
در مرحله‌ی iام، $S'_i = S_{i-1} \wedge (x_i \vee x_i \vee x_i)$ و $S''_i = S_{i-1} \wedge (\neg x_i \vee \neg x_i \vee \neg x_i)$ را می‌سازیم و هر کدام را به مسئله‌ی بررسی ارضاپذیری 3SAT می‌دهیم، اگربا ورودی S'_i بله داد، مقدار $x_i = \text{true}$ چاپ می‌کنیم و $S_i = S'_i$ قرار می‌دهیم و در غیراینصورت اگر با ورودی S''_i بله داد مقدار $x_i = \text{false}$ چاپ کرده و $S_i = S''_i$ چاپ می‌کنیم. همچنین اگر فرض کنیم S_{i-1} ارضاپذیراست حتما یکی از این دو حالت رخ می‌دهد که با استقرا می‌توان این را نشان داد.
بعد از پایان k مرحله به S_k می‌رسیم که ارضاپذیر است و همچنین مقداردهی کل متغیرها را چاپ کردیم، برای هر متغیر یکی از دو عبارت $(x_i \vee x_i \vee x_i)$ یا $(\neg x_i \vee \neg x_i \vee \neg x_i)$ در S_k موجود است که به صورت یکتا مقداردهی ارضاپذیر را مشخص می‌کند.
در این الگوریتم $2k+1$ بار الگوریتم قبلی را اجرا کردیم که طول ورودی آن حداکثر $n + 3k$ بوده پس زمان اجرای آن $(2k+1) * f(n+3k) \leq (2n+1) * f(4n)$ است که با فرض چندجمله‌ای بودن f چند جمله‌ای است و مسئله حل شد.

۲. مسئله‌ی کوله‌پشتی چندگانه به این صورت است که m کوله‌پشتی داریم که ظرفیت کوله‌پشتی iام برابر با c_i است. همچنین n الماس داریم که الماس iام ارزش v_i و وزن w_i دارد. می‌توانیم تعدادی از الماس‌ها را برداشته و هر کدام را داخل یکی از کوله‌پشتی‌ها قرار دهیم، اما مجموع وزن الماس‌های داخل یک کوله‌پشتی نباید از ظرفیت آن کوله‌پشتی بیشتر شود.
ورودی ظرفیت کوله‌پشتی‌ها، مشخصات الماس‌ها و عدد k است و می‌خواهیم ببینیم آیا می‌توانیم حداقل با مجموع ارزش k الماس داخل کوله‌پشتی‌ها جا دهیم.

(آ) برای حالت $m = 1$ (یک کوله‌پشتی) ثابت کنید مسئله‌ی کوله‌پشتی ان‌پی_تمام است.

راهنمایی برای پاسخ: اولا باید نشان دهیم که این مسئله NP است که به سادگی ممکن است، چراکه اگر لیست شماره الماس‌هایی که باید آن‌ها را در کوله‌پشتی قرار دهیم داشته‌باشیم، به سادگی می‌توان بررسی کرد که آیا وزن الماس‌ها از وزن کوله پشتی کمترمساوی و جمع ارزششان از k بیشتر مساوی است یا خیر.

حال مسئله‌ی $\mathsf{Subset-Sum}$ را به این مسئله کاهش چندجمله‌ای می‌دهیم و چون $\mathsf{Subset-Sum}$ ان‌پی_تمام است پس مسئله‌ی کوله‌پشتی نیز ان‌پی تمام است.

ورودی مسئله‌ی $\mathsf{Subset-Sum}$ یک آرایه از n عدد است و عدد t است و پرسش این است که آیا زیرآرایه‌ای با مجموع t دارد یا خیر.

کافی است با داشتن ورودی مسئله‌ی $\mathsf{Subset-Sum}$، n الماس درنظر بگیریم که وزن و ارزش الماس iام برابر با عدد iام آرایه باشد و ظرفیت کوله و عدد k را برابر با t قرار دهیم.

حال کافی‌است نشان دهیم یک ورودی برای مسئله‌ی $\mathsf{Subset-Sum}$ مقدار بله دارد اگر و تنها اگر تبدیل‌شده‌ی این ورودی به مسئله‌ی کوله‌پشتی پاسخ بله داشته باشد. که برای هر دو کافی است شماره اندیس اعضای آرایه که برای زیرآرایه انتخاب شده را متناظر با شماره اندیس الماس‌های انتخاب شده در نظر بگیریم.

(ب) برای $m = ۱$ (یک کوله‌پشتی) و با فرض $\mathbf{P} \neq \mathbf{NP}$ ثابت کنید مسئله‌ی کوله‌پشتی قویاً ان‌پی_تمام نیست.

راهنمایی برای پاسخ: ابتدا ثابت می‌کنیم این مسئله الگوریتم چندجمله‌ای دارد، سپس با برهان خلف نشان می‌دهیم که ان‌پی_تمام نیست. الگوریتم چندجمله‌ای این پرسش در مبحث برنامه‌نویسی پویا در کلاس مطرح شده‌است پس می‌دانیم که الگوریتم چندجمله‌ای دارد. همچنین اگر ان‌پی_تمام باشد همه‌ی الگوریتم‌های ان‌پی به آن کاهش می‌یابند و چون این مسئله راه حل چند جمله‌ای دارد آن‌ها نیز راه حل چندجمله‌ای خواهند داشت و در نتیجه $\mathbf{NP} \subseteq \mathbf{P}$ ، اما هر مسئله در کلاس $\mathbf{P}$ خود $\mathbf{NP}$ نیز است و در واقع $\mathbf{P} \subseteq \mathbf{NP}$ و در نتیجه $\mathbf{P} = \mathbf{NP}$ که این با فرض مسئله در تناقض است.

(ج) ثابت کنید مسئله‌ی کوله‌پشتی چندگانه قویاً ان‌پی_تمام است.

راهنمایی برای پاسخ: اولا می‌دانیم مسئله‌ی کوله‌پشتی چندگانه ان‌پی تمام است، کافی است ثابت کنیم مسئله‌ی کوله‌پشتی چندگانه با ورودی یکانی ان‌پی تمام است. برای اینکار کافی است یکی از الگوریتم‌هایی که می‌دانیم ان‌پی_تمام است را به آن کاهش چندجمله‌ای دهیم، مسئله‌ی $\mathsf{Partition}-3$ با ورودی یکانی را در نظر می‌گیریم. ورودی مسئله‌ی $\mathsf{Partition}-3$، یک آرایه از اعداد مثل $a_۱, ..., a_{۳n}$ است که $a_i \in (\frac{\sum_{i=۱}^{۳n} a_i}{۴n}, \frac{\sum_{i=۱}^{۳n} a_i}{۲n})$ و می‌خواهیم ببینیم آیا این آرایه را می‌توانیم به n دسته‌ی ۳ عضوی افراز کنیم که مجموع اعداد هر دسته باهم برابر شود یا خیر،
برای اینکار کافی‌است n کوله‌پشتی با ظرفیت $\frac{\sum_{i=۱}^{۳n} a_i}{n}$ در نظر گرفته و به‌ازای عدد iام، الماسی با وزن a_i و ارزش a_i قرار دهیم و مقدار $k = \sum_{i=۱}^{۳n} a_i$ قرار دهیم. ساختن این ورودی به صورت یکانی چندجمله‌ای است چراکه ورودی مسئله‌ی $\mathsf{Partition}-3$ نیز یکانی در نظر گرفتیم. حال کافی است نشان دهید یک ورودی پاسخ بله برای مسئله‌ی $\mathsf{Partition}-3$ می‌گیرد اگر و تنها اگر تبدیل آن به ورودی مسئله‌ی کوله‌پشتی چندگانه پاسخ بله در مسئله‌ی کوله‌پشتی چندگانه دریافت کند.

موفّق باشید.