طراحی الگوریتم

تمرین ششم - NP

آوا میرمحمدمهدی

تاریخ تحویل: ۱۴۰۳/۱۰/۲۳

در حل سوالات می‌توانید از NP-Complete بودن مسائل Independent Set و Vertex Cover علاوه بر مسائل اسلایدهای درس استفاده کنید.

۱. بازی باینری ۱۵ نمره

دو مساله‌ی A و B را در نظر بگیرید که هر کدام امکان برنده شدن یا نشدن در بازی گفته شده را تصمیم‌گیری می‌کنند. A را در زمان چندجمله‌ای به B کاهش دهید.

A: در این بازی تعدادی مهره داریم که پشت و روی آنها با رنگ‌های قرمز و آبی مشخص شده است و بر روی هر طرف آن کلمه‌ای با استفاده از الفبای انگلیسی نوشته شده است. (کلمه می‌تواند شامل یک حرف یا تعدادی حروف باشد و کلماتی که در پشت و روی یک مهره نوشته شده‌اند لزوما باهم یکسان نیستند.) همچنین از هر نوع مهره، هر تعداد که بخواهیم موجود است و اجازه داریم در زمان انتخاب مهره، پشت و روی آن را ببینیم. برنده در این بازی فردی است که بتواند تعداد متناهی از این مهره‌ها را در کنار هم قرار دهد به طوری که دنباله حروفی که از کنار هم قرار گرفتن لغات پشت مهره‌ها تشکیل می‌شود با دنباله حروفی که از کنار هم قرار گرفتن لغات روی مهره‌ها ایجاد می‌شود برابر باشد. واضح است تضمین نمی‌شود که برنده شدن در تمامی حالات ممکن باشد.

برای مثال اگر در $\frac{ab}{a}$ کلمه بالایی را کلمه‌ای که در روی مهره و کلمه‌ی پایینی را کلمه‌ای که در پشت مهره نوشته شده در نظر بگیریم، با چیدن مهره‌ها به صورت زیر برنده بازی خواهیم شد:

$$\frac{a}{ab}, \frac{bc}{ca}, \frac{a}{ab}, \frac{abc}{c}$$

B: این بازی دقیقا مانند بازی A است با این تفاوت که در تشکیل کلمات به جای حروف الفبای انگلیسی از اعداد باینری استفاده می‌شود.

برای مثال با چیدن مهره‌ها به صورت زیر برنده بازی خواهیم شد:

$$\frac{01}{011}, \frac{11}{10}, \frac{00}{01}, \frac{11}{1}$$

۲. رئوس تنها ۱۵ نمره

زیرمجموعه L از رئوس در یک گراف غیرجهت‌دار را «تنها» می‌نامیم اگر هر راس در L حداکثر با یک راس دیگر در L مجاور باشد. ثابت کنید مسالهٔ مشخص کردن اینکه یک گراف، زیرمجموعه‌ای «تنها» به اندازه k دارد یا خیر، متعلق به NP-Hard است.

۳. موش و پنیر ۱۵ نمره

موشی در یک شبکهٔ گراف جهت‌دار گیر افتاده است و می‌خواهد هرچه سریعتر پنیرهایی که روی نودهای این گراف است را بخورد. خانهٔ این موش روی گره s قرار دارد و گراف از L حلقه $C = \{R_1, R_2, \dots, R_L\}$ تشکیل شده است به طوری که تمامی این حلقه‌ها از گره s شروع می‌شوند و در هر گره، یک تکه پنیر وجود دارد. می‌خواهیم بدانیم آیا موش می‌تواند با انتخاب یک زیرمجموعه k عضوی از C و پیمایش آنها، تمام پنیرهای موجود را بخورد؟ (این k حلقه ممکن است از یک گره بیش از یک بار عبور کنند). ثابت کنید که این مساله NP-Complete است.

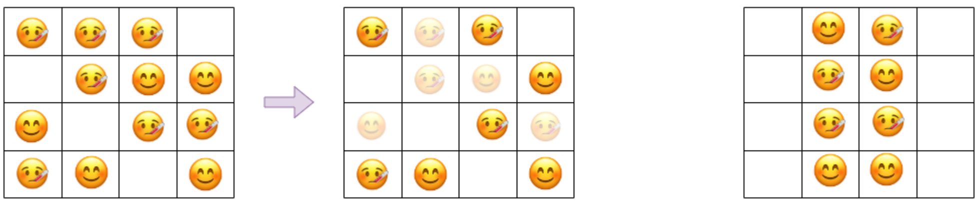
۴. بیماری بولا ۲۰ نمره

موج بیماری «بولا» از مهرماه امسال شروع شده و شیوع آن رو به افزایش است و متاسفانه تعداد زیادی از دانشجویان به آن مبتلا شده‌اند. کلاسی با n ردیف داریم به طوری که در هر ردیف m صندلی وجود دارد و دانشجویان مبتلا و سالم در برخی از این صندلی‌ها نشسته‌اند. می‌دانیم انتشار این ویروس عجیب در این کلاس در صورتی کنترل می‌شود که بتوان تعدادی از دانشجویان را به گونه‌ای از کلاس خارج کرد که دو شرط زیر برقرار باشد:

- در هر ردیف از کلاس حداقل یک نفر نشسته باشد. (چه فرد مبتلا و چه فرد سالم تفاوتی ندارد.)
- در هیچ ستونی دو نوع دانشجوی مبتلا و سالم وجود نداشته باشند.

ثابت کنید فهمیدن اینکه می‌توان بیماری بولا را با داشتن یک کلاس به همراه دانشجویانش کنترل کرد یا خیر، در دسته مسائل NP-Hard قرار دارد. (قطعا برای برخی حالات قرارگیری دانشجویان نمی‌توان این بیماری را کنترل کرد.)

برای مثال در شکل زیر در حالت (۱) با حذف دانشجویان نشان داده شده می‌توان بیماری را کنترل کرد ولی در حالت (۲) این کار امکان‌پذیر نیست.



(1) (2)

۵. زیرگراف کامل ۲۰ نمره

ثابت کنید اگر G یک گراف بدون جهت باشد، تعیین اینکه دارای یک زیرگراف کامل با حداقل $\lceil \frac{m}{2} \rceil$ گره است یا خیر، یک مسالهٔ NP-Complete است. (m تعداد رئوس گراف G است.)

۶. درست یا نادرست　　　　۲۰ نمره

فرض کنید مساله A در دسته‌ی NP-Complete قرار دارد و در زمان چندجمله‌ای قابل کاهش به مساله‌ی B است؛ همچنین می‌توان مساله‌ی C را در زمان چندجمله‌ای به A کاهش داد. درستی یا نادرستی عبارات زیر را با ذکر دلیل مشخص کنید.

الف) اگر بتوان مساله‌ی C را به طور قطعی در زمان چندجمله‌ای حل کرد، آنگاه $P = NP$ خواهد بود.

ب) اگر در آینده ثابت شود که نمی‌توان مساله‌ی B را با الگوریتمی چندجمله‌ای حل کرد، آنگاه ثابت می‌شود که $P \neq NP$ خواهد بود.

ج) اگر اثبات شود که $P \neq NP$ آنگاه قطعا نمی‌توان مساله‌ی A را در زمان چندجمله‌ای حل کرد.

د) اگر یک راه حل با پیچیدگی زمانی $\mathcal{O}(n^2)$ برای مساله‌ی B وجود داشته باشد، آنگاه می‌توان مساله‌ی A را نیز با الگوریتمی با پیچیدگی زمانی $\mathcal{O}(n^2)$ حل کرد.