طراحی الگوریتم‌ها

بهار ۱۴۰۰

مدرس: مسعود صدیقین

دانشکده‌ی مهندسی کامپیوتر

یادآوری جلسه بیست‌وپنجم | پیچیدگی محاسباتی | گردآورنده: سارا آذرنوش ــ زهرا فاضل

در جلسه قبل، در ادامه بحث پیچیدگی محاسباتی با مسائل مجموع زیرمجحوعه‌ها، کوله‌پشتی و برنامه‌نویسی صحیح و کلاس‌های مختلف مسائل آشنا شدیم.

مسئله مجموع زیرمجموعه

تعریف ۱ (مجموع زیرمجموعه). n عدد طبیعی $w_1, w_2, \ldots, w_n$ داده‌شده‌است. آیا یک زیرمجموعه از این اعداد وجود دارد که مجموع آن‌ها دقیقا برابر با w شود؟

قضیه ۱. مسئله ۳-SAT به مسئله مجموع زیرمجموعه کاهش می‌یابد.

اثبات. فرض کنیم n متغیر و m عبارت در مسئله ۳-SAT داریم. برای تبدیل مسئله ۳-SAT به مسئله مجموع زیرمجموعه به شکل زیر عمل می‌کنیم:

۱. یک جدول با $n+m$ ستون در نظر می‌گیریم که n ستون اول برای متغیرها و m ستون بعدی برای عبارت‌هاست.

۲. به ازای هر متغیر x_i دو سطر x_i و $\overline{x_i}$ را به جدول اضافه می‌کنیم. در هر دو سطر مقدار خانه iام برابر ۱ و مقدار سایر خانه‌ها در ستون n اول ۰ است. در سطر x_i به ازای هر $n+1 \leq j \leq n+m$ مقدار خانه j ۱ است اگر x_i در عبارت $j-n$ام ظاهر شده‌باشد. همین کار را برای سطر $\overline{x_i}$ نیز انجام می‌دهیم.

۳. $2m$ سطر dummy به جدول اضافه می‌کنیم. به ازای هر $1 \leq i \leq m$ خانه $n+i$ام سطر $2i-1$ام dummy برابر با ۱ و همان خانه در سطر $2i$ام dummy برابر با ۲ است. سایر خانه‌ها ۰ اند.

۴. عدد s را از مجموع اعداد جدول به گونه‌ای می‌سازیم که رقم اول n آن ۱ و m رقم بعدی ۴ باشد. یعنی بتوان در هر ستون از n ستون اول زیرمجموعه‌ای از خانه‌ها را انتخاب کرد که مجموع آن ۱ باشد و در هر ستون از m ستون بعدی بتوان زیرمجموعه‌ای از خانه‌ها را انتخاب کرد که مجموع آن‌ها ۴ باشد.

۵. اگر s وجود داشته‌باشد، مسئله ۳-SAT قابل ارضاست.

$$\Phi = (\overline{x_1} \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x_2} \vee x_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_3})$$

	x_1	x_2	x_3	c_3	c_2	c_3
x_1	1	0	0	0	1	0
$\overline{x_1}$	**1**	0	0	**1**	0	**1**
x_2	0	1	0	1	0	0
$\overline{x_2}$	0	**1**	0	0	**1**	**1**
x_3	0	0	**1**	1	**1**	0
$\overline{x_3}$	0	0	1	0	0	1
dummy	0	0	0	**1**	0	0
	0	0	0	**2**	0	0
	0	0	0	0	1	0
	0	0	0	0	**2**	0
	0	0	0	0	0	1
	0	0	0	0	0	**2**
s	**1**	**1**	**1**	**4**	**4**	**4**

□

مسئله کوله‌پشتی

تعریف ۲ (مسئله کوله‌پشتی). یک مجموعه از n داریم که شی i ام حجم v_i و ارزش w_i دارد. آیا می‌توان یک زیرمجموعه از اشیا با ارزش حداقل W و حجم حداکثر V انتخاب کرد؟

قضیه ۲. مسئله مجموع زیرمجموعه به مسئله کوله‌پشتی کاهش می‌یابد.

اثبات. وزن و ارزش هر عدد را برابر با خود آن عدد و W و V را برابر با مجموع خواسته‌شده قرار می‌دهیم. فرض کنیم مجموعه اعداد $\{a_1, a_2, \ldots, a_n\}$ و مجموع خواسته‌شده S باشد و x_i برابر با ۱ باشد، اگر a_i در مجموعه باشد. $\square$

$$\left.\begin{array}{l} \sum_{i=1}^{n} x_i v_i = \sum_{i=1}^{n} x_i a_i \leq V = S \\ \sum_{i=1}^{n} x_i w_i = \sum_{i=1}^{n} x_i a_i \geq W = S \end{array}\right\} \implies \sum_{i=1}^{n} x_i a_i = S$$

مسئله برنامه‌نویسی صحیح

تعریف ۳ (مسئله برنامه‌نویسی صحیح). یک برنامه نویسی خطی با شرط صحیح بودن متغیرها است.

قضیه ۳. مسئله کوله‌پشتی به مسئله برنامه‌نویسی صحیح کاهش می‌یابد.

اثبات. مسئله کوله پشتی به شکل روبه‌رو به یک مسئله برنامه‌نویسی صحیح تبدیل می‌شود. $\square$

$$\begin{array}{ll} \max & \sum_{i=1}^{n} x_i w_i \\ \text{s.t.} & \sum_{i=1}^{n} x_i v_i \geq V, \\ & \forall 1 \leq i \leq m : x_i \in \{0, 1\} \end{array}$$

دسته‌بندی مسائل

تعریف ۴ (دسته P). مجموعه همه مسائل تصمیم‌گیری که برای آن‌ها الگوریتم چندجمله‌ای وجود دارد.

تعریف ۵ (دسته NP). مجموعه همه مسائل تصمیم‌گیری که برای آن‌ها یک تصدیق چندجمله‌ای وجود دارد. یعنی می‌توان درستی یک جواب را در زمان چندجمله‌ای بررسی کرد.

تعریف ۶ (دسته NP Complete). یک مسئله متعلق به دسته NP در دسته NP Complete نیز هست اگر هر مسئله دسته NP را بتوان به آن کاهش داد.

تعریف ۷ (دسته NP Hard). یک مسئله در دسته NP Hard هست اگر هر مسئله دسته NP را بتوان به آن کاهش داد.

قضیه ۴ (قضیه کوک). مسئله SAT یک مسئله NP Complete است.

می‌توان نشان داد مسئله SAT به مسئله ۳-SAT کاهش می‌یابد. بنابراین این دو مسئله معادل بوده و مسئله ۳-SAT نیز NP Complete است.