فصل نهم

مسائل رام‌نشدنی

مقدمه

همانطور که تا حالا متوجه شدید راه‌حل‌های ارائه شده برای مسائل در حالت کلی غالباً به دو صورت ظاهر می‌شوند. اولی الگوریتمهایی که پیچیدگی زمانی آنها حداکثر چندجمله‌ای می‌باشد و دومی مسائلی که الگوریتمهای ارائه شده برای آنها از درجه نمایی می‌باشد. الگوریتمهای ارائه شده برای مسائلی از قبیل مرتب‌سازی، ضرب ماتریس‌ها، درخت پوشای کمینه و غیره از نوع دسته اول می‌باشند و الگوریتمهای ارائه شده برای مسائلی از قبیل رنگ‌آمیزی گراف‌ها، کوله‌پشتی ۰/۱، فروشنده دوره‌گرد و غیره از نوع دوم می‌باشند. و همانطور که ملاحظه کردید الگوریتمهایی که پیچیدگی زمانی آنها نمائی می‌باشد در عمل کاربرد خاصی ندارند و برای حل یک مثال نه چندان بزرگ ممکن است ماه‌ها یا سال‌ها زمان نیاز داشته باشند.

همانطور که ملاحظه کردید، در فصل‌های قبل الگوریتمهای ارائه شده برای حل مسائلی از قبیل کوله‌پشتی، مسئله n وزیر و فروشنده دوره‌گرد از مرتبه نمائی بودند. و در عین حال وجود الگوریتمهایی با درجه پایین‌تر رد نشد. در این فصل قصد داریم در مورد چنین الگوریتمهایی بطور مختصر بحث کنیم.

سناریوی فوق دقیقاً همان چیزی است که دانشمندان علوم کامپیوتر با موفقیت در طی ۲۵ سال اخیراً انجام داده‌اند. نشان داده‌اند که مسأله فروشنده دوره‌گرد و هزاران مسأله دیگر هم‌ارز هستند. چرا که با داشتن الگوریتمی کارآمد برای یکی از آنها، برای

همه آنها الگوریتمی کارآمد خواهیم داشت. چنین الگوریتمی هرگز ابداع نشده است ولی غیرممکن بودن آن نیز هنوز به اثبات نرسیده است، این‌گونه مسائل جالب را «NP کامل» می‌گویند (به مسائلی که نوشتن یک الگوریتم کارآمد برای آنها غیرممکن است مسائل رام‌نشدنی (Intractable) می‌گویند)

۱-۹ مسائل رام‌نشدنی

همان‌گونه که در مقوله قبل اشاره شد(مسائلی که نتوان برای آنها الگوریتمی با مرتبه زمانی چندجمله‌ای پیدا کرد مسائل رام‌نشدنی نامیده می‌شود. الگوریتمهایی با مرتبه زمانی ۲^n، ۳^n، $n!$ یا هر الگوریتمی که مرتبه زمانی آن غیر چندجمله‌ای باشد (یعنی نمایی باشد) را مسائل رام‌نشدنی می‌نامند) تعداد اعمال لازم برای اجرای الگوریتمهای رام‌نشدنی با رشد n به شدت رشد می‌کنند و تا حدی می‌رسند که حتی حل آنها توسط ابرکامپیوترها در مدت زمان معقول ناممکن می‌شود.

قابل ذکر است مرتبه‌های زمانی نمائی همیشه نامطلوب نیستند و برای مسائلی با اندازه‌های کوچکتر ممکن است مرتبه‌های زمانی نمائی کارایی مطلوبتری داشته باشند.

در حالت کلی، در برخورد با مسائل به سه گروه از راه‌حل‌ها با مرتبه‌های زمانی زیر می‌رسیم:

۱. مسائلی که الگوریتمهای زمانی چندجمله‌ای برای آن یافت شده است.
۲. مسائلی که رام‌نشدنی بودن آنها ثابت شده است.
۳. مسائلی که رام‌نشدنی بودن آنها اثبات نشده است ولی هیچ الگوریتم زمانی چندجمله‌ای هم برای آنها پیدا نشده است.

۲-۹ مسائلی که الگوریتمهای زمانی چندجمله‌ای برای آنها پیدا شده است.

هر مسأله‌ای که یک الگوریتم زمانی چندجمله‌ای برای آن پیدا شده است در این گروه قرار دارد. به عنوان مثال برای مرتب‌سازی الگوریتمهای $O(n \operatorname{Log} n)$، برای جستجو در یک آرایه مرتب، یک الگوریتم $O(\operatorname{Log} n)$، برای ضرب ماتریس‌ها یک الگوریتم

$O(n^{2.38})$ ضرب زنجیره‌ای ماتریس‌ها یک الگوریتم $O(n^{3})$ و غیره الگوریتمهایی از مرتبه چندجمله‌ای پیدا شده است.

در کل مسائلی که برای آنها الگوریتمهایی از مرتبه چندجمله‌ای ارائه شده باشد مسائل بغرنجی نیستند و حل آنها ساده می‌باشد.

۳-۹ مسائلی که رام‌نشدنی بودن آنها ثابت شده است.

مسأله تعیین کلیه مدارهای هامیلتونی که در این مسأله تعداد مدارها $(n-1)!$ می‌باشد، را در نظر بگیرید. اثبات رام‌نشدنی بودن این گونه مسائل راحت است. در این مسأله با توجه به عبارت محاسبه تعداد مدارها می‌توان دریافت که پیچیدگی زمانی این مسأله $N!$ می‌باشد که جزء مسائل رام‌نشدنی می‌باشد.

همه مسائلی که تا این تاریخ رام‌نشدنی بودن آنها اثبات شده است، نبودن آنها در مجموعه NP نیز اثبات شده است. قابل ذکر است که رام‌نشدنی بودن تعداد نسبتاً کمی از مسائل اثبات شده است.

۴-۹ مسائلی که رام‌نشدنی بودن آنها ثابت نشده است ولی تاکنون هیچ الگوریتم زمانی چندجمله‌ای برای آنها پیدا نشده است.

این گروه شامل تمام مسائلی می‌شود که هیچ الگوریتم زمانی چندجمله‌ای برای آنها پیدا نشده است ولی هنوز کسی غیرممکن بودن این چنین الگوریتمی را اثبات نکرده است. مسائل بسیاری از این نوع وجود دارد. برای مثال، اگر مسائل را طوری بیان کنیم که تنها یک حل مورد نیاز باشد(مسأله کوله‌پشتی صفر و یک، مسأله فروشنده دوره‌گرد، مسأله حاصل‌جمع زیرمجموعه‌ها، مسأله رنگ‌آمیزی m به ازای $m \geq 3$، مسأله مدارهای هامیلتونی همگی در این گروه قرار دارند) برای این مسائل، الگوریتمهای انشعاب و تحدید، الگوریتمهای عقبگرد وجود دارد که برای بسیاری از نمونه‌ها بازدهی دارند. ولی الگوریتمهای ارائه شده با این روش‌ها برای این مسائل از مرتبه نمائی می‌باشد و همانطور که اشاره شد احتمال وجود الگوریتمهای کاراتر رد نمی‌شود.

۹-۵ نظریه NP

نخست برای ورود به دنیای بررسی مسائل از نظر نوع الگوریتمهای قابل ارائه برای حل آنها، مسائل تصمیم‌گیری را تعریف می‌کنیم(هر مسأله‌ای که جواب آن بلی یا خیر باشد یک مسأله تصمیم‌گیری است.)

کلاسهای P، NP، NP-Complete و NP-hard از مسائل، همه در چارچوب مسائل تصمیم‌گیری تعریف و بررسی می‌شوند. که در ادامه بحث آنها را تعریف خواهیم کرد.

در کل تقریباً تمام مسائل بهینه‌سازی که تا اینجا بررسی کردیم از نوع مسائل تصمیم‌گیری می‌باشند. به عبارت دیگر می‌توان مسائل مذکور را بصورت مسائل تصمیم‌گیری بیان کرد. برای نمونه به مثال زیر توجه کنید.

مثال ۹ .۱ : مسئله کوله‌پشتی ۰/۱ را در نظر بگیرید. در این مسئله همانطور که قبلاً اشاره کردیم، هدف انتخاب تعدادی شیء با وزن و ارزش مشخص بود بطوریکه ارزش اشیاء انتخاب شده بیشترین مقدار بوده و مجموع وزن اشیاء مذکور از ظرفیت کوله‌پشتی بیشتر نباشد.

همانطور که می‌دانید مسئله مذکور یک مسئله بهینه‌سازی می‌باشد. حال آن را به شکل یک مسئله تصمیم‌گیری بیان می‌کنیم.

مسئله تصمیم‌گیری کوله‌پشتی ۰/۱ بدین صورت می‌توان بیان کرد:

برای ارزش یا سود مفروض P، آیا می‌توان بیشترین سود یا ارزش، که از انتخاب اشیاء و قرار دادن در کوله‌پشتی حاصل می‌شود، تعیین نمود. بطوریکه وزن اشیاء انتخاب شده، از ظرفیت کوله‌پشتی تجاوز نکند و ارزش کل حداقل P باشد، آیا چنین انتخابی ممکن است یا خیر؟

همانطور که ملاحظه می‌کنید این همان مسئله بهینه‌سازی بالا بوده بعلاوه اینکه شرط P نیز به آن اضافه شده است.

همانطور که ملاحظه می‌کنید می‌توان مسائلی از این قبیل را به صورت مسائل تصمیم‌گیری بیان نمود.

۹-۶ کلاس P (Polynomial)

مسائلی که برای حل آنها الگوریتم یا الگوریتمهایی با مرتبه زمانی چندجمله‌ای یافت شده است کلاس الگوریتمهای قطعی را تشکیل می‌دهند. این کلاس را با P که مخفف Polynomial یا چندجمله‌ای می‌باشد، نمایش می‌دهند. پس P عبارت است از مجموعه تمامی مسائل تصمیم‌گیری که توسط الگوریتمهای زمانی چندجمله‌ای قابل حل هستند.

الگوریتمهایی که برای کامپیوترهای شخصی رایج طراحی می‌گردند الگوریتمهای قطعی نامیده می‌شوند و در آنها نتیجه هر عمل کاملاً معیّن و قطعی است. به عنوان مثال، جستجوی دودوئی، الگوریتمهای مرتب‌سازی، جستجو و غیره در این کلاس می‌باشند.

کدام مسائل به مجموعه P تعلق دارند؟ آیا ممکن است یک مسأله تصمیم‌گیری که برای آن هیچ الگوریتم زمانی چندجمله‌ای یافت نشده است به P تعلّق داشته باشد؟ آیا مسأله فروشنده دوره‌گرد به مجموعه P تعلّق دارد؟ اگرچه تاکنون کسی موفق به یافتن الگوریتم زمانی چندجمله‌ای برای این مسأله نشده است، از طرف دیگر، کسی هم تاکنون اثبات نکرده است که حل آن با یک الگوریتم زمانی چندجمله‌ای غیرممکن است. بنابراین این امکان وجود دارد که مسئله مذکور در مجموعه P باشد. برای آنکه بدانیم یک مسأله تصمیم‌گیری در P نیست باید ثابت کنیم که طراحی یک الگوریتم زمانی چندجمله‌ای برای آن غیرممکن است. این کار برای مسأله فروشنده دوره‌گرد انجام نشده است.

۹-۷ کلاس NP (Nondeterministic Polynomial)

برای مسائل کلاس NP باید کامپیوتر علاوه بر توانایی اجرای دستورهای معیّن و قطعی، قادر باشد برخی دستورات نامعین را نیز اجرا کند. یک دستور نامعین دستوری است که نتیجه اجرای آن از قبل قابل پیش‌بینی نباشد. برای مثال فرض کنید دستوری داشته باشیم که بخواهد از بین ۱۰۰ شی یکی را انتخاب کند. قبل از اجرای چنین دستوری نمی‌توان پیش‌بینی کرد که دقیقاً کدامیک ازاشیاء انتخاب خواهد شد. چنین کامپیوتری را

یک کامپیوتر نامعیّن (Nondeterministic) می‌نامند. الگوریتمهایی که برای یک کامپیوتر نامعین طراحی می‌شوند الگوریتمهای نامعین نامیده می‌شوند.

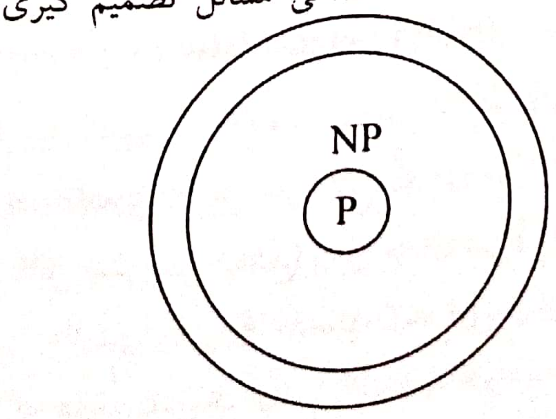
الگوریتمهای نامعین علاوه بر دستورهایی که برای بیان الگوریتم معیّن وجود دارد، باید دستورات دیگر را نیز اضافه کنند. معمولاً دستورهای زیر به الگوریتمهای معین اضافه می‌شود تا الگوریتم به نامعین تبدیل شود:

الف) تابعی که یکی از عناصر مجموعه S را به دلخواه انتخاب کند (اولین مرحله، مرحله حدس زدن نامیده می‌شود). ممکن خروجی حاصل از این مرحله بی‌ربط باشد.

ب) حدس انتخاب شده و مجموعه S ورودی این تابع می‌باشد. خروجی این تابع، پایان موفق یا ناموفق عملیات الگوریتم را اعلام می‌کند (این مرحله، مرحله تصدیق نامیده می‌شود).

در بیان یک الگوریتم نامعین لزومی ندارد از همه دستورها و تابع‌های ذکر شده استفاده شود. بنابراین هر الگوریتم معیّن توسط یک کامپیوتر نامعیّن قابل اجرا است. پس $P \subseteq NP$.

محققین زیادی سعی کرده‌اند ثابت کنند P=NP می‌باشد. اگر این مسأله ثابت شود مفهوم آن این است که هر مسأله‌ای که برای آن الگوریتم نامعین با مرتبه زمانی چندجمله‌ای وجود دارد می‌توان برایش یک الگوریتم معین با مرتبه زمانی چندجمله‌ای پیدا کرد. در وضعیّت کنونی شکل ۱-۹ رابطه کلاسهای P و NP را نشان می‌دهد.



شکل ۱-۹ مجموعه تمامی مسائل تصمیم‌گیری

حال با چند تعریف بحث را تکمیل می‌کنیم.

تعریف: الگوریتم غیر قطعی با زمان چندجمله‌ای، الگوریتم غیر قطعی است که، مرحله تصدیق آن دارای زمان چند جمله‌ای باشد.

بنابراین با توجه به تعریف بالا مسائل از نوع NP را بصورت ذیل تعریف می‌کنیم.

تعریف: NP مجموعه تمامی مسائل تصمیم‌گیری است که توسط الگوریتمهای غیرقطعی با زمان چندجمله‌ای قابل حل باشند. NP نشان‌دهنده «چندجمله‌ای غیرقطعی» است. برای آنکه یک مسأله تصمیم‌گیری در NP باشد، باید الگوریتمی وجود داشته باشد که عمل تصدیق را در زمان چندجمله‌ای انجام دهد. تأکید بر این نکته ضروری است که این بدان معنا نیست که الزاماً الگوریتمی با زمان چندجمله‌ای برای حل مسأله داریم. در حال حاضر چنین الگوریتمی برای مسأله فروشنده دوره‌گرد نداریم.

۹-۸ خلاصه فصل

۱. مسائلی که نوشتن یک الگوریتم کارآمد برای آنها غیرممکن است مسائل رام‌نشدنی می‌گویند.

۲. مسائلی که الگوریتم کارا (چندجمله‌ای) برای آنها ابداع نشده است ولی غیرممکن بودن آن نیز هنوز به اثبات نرسیده را مسائل NP کامل می‌گویند.

۳. مسأله فروشنده دوره‌گرد، مسأله n وزیر، مسأله رنگ‌آمیزی گراف و مسأله کوله‌پشتی جزو مسائلی هستند که تا حال نتوانسته‌اند الگوریتمی با مرتبه زمانی چندجمله‌ای برای آنها پیدا کنند.

٤. الگوریتمهایی با مرتبه زمانی ۲^n، ۳^n و $n!$ ... را مسائل رام‌نشدنی می‌نامند.

۹-۹ تمرینات

۱. سه مسأله NP پیدا کرده و به دقت تشریح کنید.

۲. برای مرتب‌سازی داده‌های آرایه‌ای از اعداد یک الگوریتم نامعیّن طراحی کنید.

۳. یک الگوریتم مرتبه زمانی چندجمله‌ای بنویسید که چک کند آیا یک گراف بدون جهت دارای مدارهای هامیلتونی هست یا خیر. فرض کنید این گراف هیچ رأسی با درجه بیشتر از ۲ ندارد.

٤. تابع حدس اولیه برای مسائل NP بنویسید و آنرا تحلیل نمائید.

٥. مسائل فروشنده دوره‌گرد، رنگ‌آمیزی گراف و مجموع زیرمجموعه‌ها را به شکل مسائل تصمیم‌گیری ارائه دهید.

٦. تابع تصدیق‌پذیری را نوشته و تحلیل نمائید.

۷. مسائل NP-hard را تعریف نموده، سپس رابطه آن را با مسائل NP مشخص نمائید.

۸. مسائل NP-complete را تعریف نموده، سپس رابطه آن را با مسائل NP مشخص نمائید.

۹. رابطه بین مسائل از نوع P ، NP-hard ، NP-complete را مشخص نمائید.