

پاسخنامه آزمون الگوریتم‌های پیشرفته

استاد: دکتر سید علی رضوی ابراهیمی
دانشجو: مریم سادات موردگر - ۹۹۰۱۹۰۴۲۶

تابستان ۱۴۰۰



سؤالات میان ترم

۱- در معنی و مفهوم علمی الگوریتم بحث نمایید. درس الگوریتم و الگوریتم‌های مورد مطالعه برای مهندسی نرم افزار چه کاربرد و فایده‌ای دارند؟

الگوریتم مجموعه متناهی از دستورالعمل‌هاست که به ترتیب خاصی اجرا می‌شوند و مسئله را حل می‌کنند به عبارت دیگر یک الگوریتم، روش گام‌به‌گام برای حل مسئله است.

الگوریتم روش حل سیستماتیک مسئله است قبل از اینکه حل مسئله رخ دهد مهم است اول بدانیم که طبیعت مشکل چیست اگر مسئله را اشتباه درک کرده باشید تلاش‌های شما برای حل مسئله نیز ممکن است اشتباه یا بیهوده باشد. الگوریتم‌ها دارای خصوصیتی هستند مانند: دارای نقطه شروع و پایان هستند ورودی و خروجی‌ها مشخص باشد و مراحل کار شفاف و بدون ابهام تعیین شود، تکرارپذیرند. از لحاظ عملی هم بهره‌وری داشته باشند و بتوان با هزینه و زمان مقول یعنی سرعت بیشتر و زمان کمتر، مسئله را حل کرد.

تفاوت متد و الگوریتم: متد شیوه یا روش نگاه به مسئله است در متد کلیات یا جهت حل مسئله معین می‌شود و جزییات و مراحل بیان نمی‌شود در اصل یک نسخه عمومی‌تر و کلی‌تر از الگوریتم است و در آن ابهام وجود دارد ولی الگوریتم گام‌های حل مسئله به‌طور سیستماتیک و دقیق‌تر است و در آن هیچ ابهامی وجود ندارد.

ما معمولاً برای حل مشکلات به دنبال ساده‌ترین و سریع‌ترین راه‌حل‌ها هستیم. سال‌ها است که علم با یافتن پاسخ سؤالات خود و استفاده از آن‌ها در پیشامدهایی که الگوی تکراری دارند، اهداف خود را پیش می‌برد و سریع‌تر از انتظار ما رازهای طبیعت را از دل آن بیرون می‌کشد. راه‌حلهایی که تست‌شده و مطمئن هستند و می‌توانند سؤالاتی با مفاهیم یکسان را حل کنند، الگوریتم نامیده می‌شوند. این روش برای نخستین بار توسط خوارزمی مورد توجه قرار گرفت و نام الگوریتم نیز به احترام این دانشمند و از کلمه الخوارزمی گرفته شده است.

اگر بخواهیم معنی الگوریتم را در زمینه ریاضیات و علوم کامپیوتر بررسی کنیم، می‌توان گفت الگوریتم‌ها مجموعه فرایندهایی هستند که به کمک آن‌ها می‌توان بسیاری از مسائل برنامه‌نویسی را به راحتی حل کرد. به عنوان مثال الگوریتم یک موتور جستجو را در نظر بگیرید. الگوریتم موتور جستجو گوگل به‌طور ساده این گونه ست که عبارت تایپ شده شمارا دریافت کرده و آن را در پایگاه داده‌های خود جستجو می‌کند.

الگوریتم در اصل مفهومی است که مسائل را با استفاده از دستورالعمل‌های پشت سرهم و به کمک تحلیل‌های ریاضی و منطقی مورد بررسی قرار داده و راه‌حل مناسبی برای آن ارائه می‌کند، می‌توان الگوریتم را رویه‌ای تعریف کرد که شامل تعداد متناهی از دستورالعمل‌هاست و هدف خاصی را دنبال می‌کند، یکی از مهم‌ترین اهداف در علم کامپیوتر، فرایند طراحی الگوریتم برای تولید خروجی صحیح درازای ورودی، در محدوده زمانی متناهی است، کارایی یک الگوریتم با آیت‌های مختلفی سنجیده می‌شود ولی مهم‌ترین جنبه سنجش میزان کارایی یک الگوریتم، زمان اجرای آن است یعنی شمارش تعداد زمان‌هایی که عملیات اساسی الگوریتم اجرا شده است. از دیگر جنبه‌های سنجش، فضای مورد نیاز برای اجرای آن است.

اشخاص در مشاغل گوناگون برای انجام کار روزانه خود از الگوریتم‌های ذهنی یا مکتوب خاصی برای انجام کارها یا انجام وظایف بهره می‌برند. یکی از مباحث درسی رشته‌های مهندسی کامپیوتر و فناوری اطلاعات است، یک مبحث پایه‌ای برای فعالیت در حوزه‌های هوش مصنوعی، یادگیری ماشین و علم داده به شمار می‌آید. تسلط بر مبحث تحلیل و طراحی الگوریتم‌ها به برنامه‌نویس‌ها نیز در حل بهتر مسائل برنامه‌نویسی و نوشتن کدهایی با کارایی و سرعت اجرای بالاتر و مصرف حافظه کمتر کمک قابل توجهی می‌کند. یافتن راه‌حل برای مسائل گوناگون و توانایی حل مسائل پیچیده مهندسی‌ن قادر خواهند بود که با استفاده از مباحث الگوریتم‌ها و توصیف راستی آزمایی و راه‌حل برای مسائل واقعی و مطرح روز بکار ببرند.

در علوم رایانه یک الگوریتم را روال محاسباتی خوش تعریف می‌دانند که مقدار یا مجموعه‌ای از مقادیر را به‌عنوان ورودی دریافت کرده و پس از طی چند گام محاسباتی، ورودی را به خروجی تبدیل می‌کند. به‌جز این الگوریتم را ابزاری برای حل مسائل محاسباتی نیز تعریف کرده‌اند. یک برنامه رایانه‌ای بیان یک یا چند الگوریتم با یک‌زبان برنامه‌نویسی است. معمولاً برای حل یک مسئله روش‌ها و الگوریتم‌های گوناگونی وجود دارد، یک الگوریتم ممکن است عمل موردنظر را با دستورهای مختلف در مدت‌زمان یا کار کمتر یا بیشتری نسبت به الگوریتم دیگر انجام دهد. به همین دلیل انتخاب الگوریتم مناسب و کارا اهمیت زیادی در موفق بودن و کارایی برنامه رایانه‌ای دارد. الگوریتم‌ها به‌عنوان یک علم مطرح هستند و دانشمندان آن‌ها را طراحی، تحلیل و مطالعه می‌کنند.

معمولاً برای حل یک مسئله، روش‌ها و الگوریتم‌های گوناگونی وجود دارند؛ یک الگوریتم ممکن است عمل موردنظر را با دستورهای مختلف در مدت‌زمان یا کار کمتر یا بیشتری نسبت به الگوریتم دیگر انجام دهد. به همین دلیل، انتخاب الگوریتم مناسب و کارا اهمیت زیادی در موفق بودن و کارایی برنامه رایانه‌ای دارد. الگوریتم‌ها به‌عنوان یک علم مطرح هستند و دانشمندان آن‌ها را طراحی، تحلیل، و مطالعه می‌کنند. مطالعه الگوریتم‌ها زمینه‌های متعددی را در برمی‌گیرد. در زیر به چند نمونه اشاره می‌کنیم که می‌توان آن‌ها را چرخه حیات یک الگوریتم نامید.

الف) طراحی الگوریتم‌ها: روش‌های مختلفی برای طراحی الگوریتم‌ها وجود دارد که عبارتند از روش‌های تقسیم و غلبه، روش‌های حریصانه، روش‌های برنامه‌نویسی پویا، روش‌های پسگرد و روش‌های انشعاب و تحدید.

ب) معتبر سازی یا اثبات درستی الگوریتم‌ها: بعد از طراحی باید اثبات شود که الگوریتم مزبور درست است. الگوریتمی درست است که به ازای هر ورودی مناسب خروجی صحیحی بدهد. اثبات درستی الگوریتم‌ها به اثبات قضایا در ریاضی می‌ماند و مرحله بسیار مهمی در زمینه مطالعه الگوریتم‌ها است

پ) تحلیل الگوریتم‌ها (تحلیل مقدم، ارزیابی کارایی الگوریتم‌ها): یک الگوریتم در زمان اجرا از واحد پردازش مرکزی رایانه برای اجرای دستورالعمل‌ها و از حافظه برای ذخیره‌سازی برنامه و داده‌ها استفاده می‌کند. تحلیل یک الگوریتم مشخص می‌کند که الگوریتم در زمان اجرا برای چه مدتی از CPU برای اجرای دستورالعمل (پیچیدگی زمانی) استفاده کرده و چه مقدار از حافظه (چه اصلی و چه جانبی) برای ذخیره‌سازی برنامه و داده‌ها (پیچیدگی فضایی) به‌کاربرده است. تحلیل الگوریتم بیان‌گر آن است که، یک الگوریتم به چه میزان پیچیدگی زمانی و پیچیدگی فضایی نیاز دارد.

ت) پیاده‌سازی الگوریتم‌ها: پیاده‌سازی یک الگوریتم نوشتن آن به زبان برنامه‌نویسی خاص است که معمولاً بعد از تحلیل مقدم آن صورت می‌گیرد و نام برنامه به آن اطلاق می‌شود.

ث) تست برنامه: تست یک برنامه شامل: اشکال زدایی و تحلیل مؤخر (اندازه‌گیری کارایی) است. اندازه‌گیری کارایی عبارت است از فرایند اجرای الگوریتم صحیح بر روی داده‌های نمونه‌گیری شده برای به دست آوردن زمان و حافظه مورد نیاز توسط کامپایلر. زمان اجرای یک الگوریتم به پارامترهای مختلفی بستگی دارد که از جمله می‌توان به نوع دستورالعمل‌ها (دستورالعمل‌های جمع، ضرب، نوشتن، خواندن، شرطی و...) کامپایلر مورد استفاده، زبان برنامه‌نویسی، سخت‌افزار به کار رفته و پارامتری مثل n که می‌تواند معرف تعداد ورودی‌ها و خروجی‌ها یا هر دو باشد اشاره کرد.

تحلیل الگوریتم‌ها رشته‌ای است که به بررسی کارایی الگوریتم‌ها می‌پردازد. تحلیل الگوریتم‌ها یعنی پیش‌بینی منابع مورد نیاز برای اجرای یک الگوریتم، همچون: حافظه، پهنای باند ارتباطی، سخت‌افزار، و از همه مهم‌تر، زمان. کارایی یا پیچیدگی هر الگوریتم را با تابعی نشان می‌دهند که تعداد مراحل لازم برای اجرای الگوریتم را بر حسب طول داده ورودی، یا میزان محل‌های لازم حافظه را بر حسب طول داده ورودی نشان می‌دهد.

۲- نحوه عمل نیم پاک‌کننده در شبکه‌های مرتب ساز را شرح دهید؟

راهنمایی: در مفهوم و معنی عبارت زیر دقت نمایید:

A half-cleaner ensures that every output element of the top half is at least as small as every output element of the bottom half. Moreover, both halves are bitonic, and at least one half is clean.

شبکه مرتب‌سازی یک مدل انتزاعی ریاضی شامل شبکه‌ای از سیم‌ها و واحدهای مقایسه کننده است که برای مرتب‌سازی رشته‌ای از اعداد از آن استفاده می‌شود. هر مقایسه کننده دو سیم را به هم متصل می‌کند و مقادیر را با قرار دادن مقدار کوچک‌تر به یکی از سیم‌ها و مقدار بزرگ‌تر به سیم دیگر، مرتب می‌کند. اصلی‌ترین تفاوت میان شبکه مرتب‌سازی و الگوریتم مرتب‌سازی این است که ترتیب مقایسه‌ها بدون در نظر گرفتن نتیجه مقایسه‌های قبلی، از قبل مشخص شده است. استقلال میان ترتیب مقایسه‌ها برای اجرای موازی الگوریتم‌ها مفید خواهد بود. علی‌رغم سادگی مدل آن، تئوری مرتب‌سازی شبکه‌ای بسیار پیچیده و دارای مفاهیم عمیقی است. یک شبکه مرتب ساز، یک شبکه مقایسه‌گر است که رشته‌های ورودی را دریافت کرده و آن را به صورت مرتب‌شده در خروجی نمایش می‌دهد به بیان دیگر به مداری که n ورودی را دریافت کرده و با مرتب کردن آن‌ها تعداد n خروجی را تولید می‌کند پایه و اساس شبکه‌ها و الگوریتم‌های مرتب‌سازی مبتنی بر مقایسه و جابجایی جفت کلیدها است.

شبکه مرتب ساز Bitonic :

دنباله‌ای که به صورت یکنواخت افزایش می‌یابد و سپس کاهش می‌یابد یا می‌توان آن را به صورت گردشی تغییر داد تا یکنواخت افزایش و سپس یکنواخت کاهش می‌یابد بنابراین می‌توان گفت با استفاده از مرتب‌سازی بایتونیک می‌توان یک شبکه مقایسه‌ای را به صورت صفرها و یک‌ها مرتب کرده و سپس آن را برای هر عدد دلخواه دیگر بکار برد. شبکه‌های مرتب‌سازی به دست آمده از $O(n \log(n)^2)$ مقایسه کننده تشکیل شده و تأخیری به میزان $O(\log(n)^2)$ دارند که n تعداد مواردی است که باید مرتب شوند. توالی مرتب شده یک توالی یکنواخت غیر کاهشی یا غیر افزایشی است. توالی بایتونیک یک توالی با $x_0 \leq \dots \leq x_k \geq \dots \geq x_{n-1}$ برای برخی مقادیر k ، که مقادیر آن در محدوده $0 \leq k < n$ قرار دارند، با شیفت دورانی این توالی است.

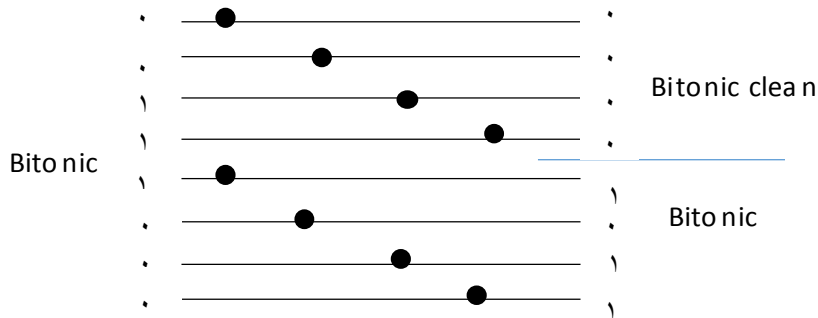
نیم پاک کننده:

در مرتب‌سازی بایتونیک عمل مرتب‌سازی در چندین مرحله انجام می‌شود به هر یک از این مراحل نیم پاک کننده گفته می‌شود. در شبکه مقایسه‌ای half-cleaner ورودی بر روی خط i با ورودی خط $i-n/2$ مقایسه می‌شود فرض بر این است که ورودی‌ها به صورت دو آهنگی یکنواخت افزایش و یا کاهشی باشند بعد از پایان مقایسه حداقل یک نیمه تمیز خواهیم داشت.

عمق هر نیم پاک کننده، برابر با ۱ خواهد بود. خط i با خط $i-n/2$ مقایسه شده و مقادیر کوچکتر را در نیمه بالایی قرار می‌دهد و نیمه پایینی بایتونیک است.

در half-cleaner حداقل یکی از نصفه‌ها تمیز باشد، در تعداد صفر و یک‌ها، یا تعداد یک‌ها و صفرها مساوی است یا تعداد یک‌ها یا تعداد صفرها کمتر است در هر حالت اگر تعداد صفرها بیشتر از نصف باشد، در واقع یک نصفه صفر خواهیم داشت اگر تعداد یک‌ها بیشتر باشد نصفه یک خواهیم داشت و یک نصفه خواهیم داشت که ترکیبی از صفر و یک است. در هر حالت half-cleaner یک نصفه تمیز خواهیم داشت و آن نصفه دیگر حتماً بایتونیک خواهد بود.

خصوصیات



اگر ورودی نیمه تمیزکننده یک توالی بایتونیک از ۰ و ۱ باشد در این صورت توالی خصوصیات زیر را برآورده می‌کند:

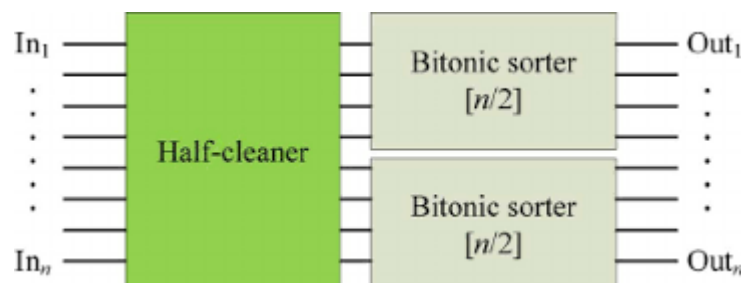
۱- هر دونیمه بالا و پایین بایتونیک هستند.

۲- هر المان در نیمه بالایی حداکثر به اندازه هر المان نیمه پایینی کوچک خواهد بود.

۳- حداقل نیمی از آن تمیز است (یا همه صفر یا یک)

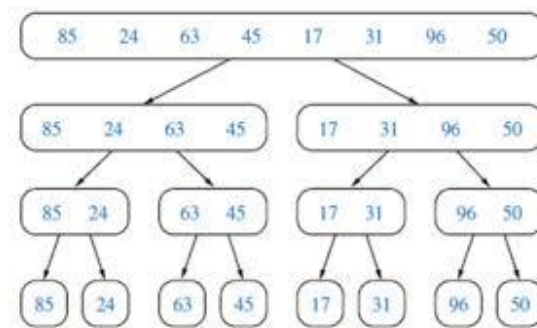
در واقع نیم پاک‌کننده بیان می‌کند که هر عنصر خروجی از نیمه بالا حداقل به اندازه هر عنصر از نیمه پایین می‌باشد و حداقل یک‌نیمه از آن تمیز می‌باشد.

در شکل زیر یک توالی دو آهنگی یکنواخت افزایشی را نشان می‌دهد. بعد از مرحله اول مرتب‌سازی، درست‌چپ، یک نیم پاک‌کننده از صفرها و سمت راست، یک توالی دو آهنگی یکنواخت کاهشی تولید خواهد شد. عمق هر نیم پاک‌کننده، برابر با یک بوده و خواهیم داشت: $D(n) = 1$



شبکه های ادغام در ساخت شبکه های مرتب سازی مورد استفاده قرار میگیرد. هر شبکه مرتب سازی که بتواند برای بدست آوردن دنباله مرتبی به طول n ، یک عنصر را با $n-1$ عنصر مرتب، ادغام نماید، حداقل دارای عمق $\log n$ خواهد بود. ضمن اینکه، هر شبکه ادغام صرفنظر از ترتیب ورودی ها به $\Omega(n \log n)$ مقایسه گر نیاز دارد. بهترین زمان برای مرتب سازی n عنصر در شبکه های مرتب سازی نیز برابر $O(\lg n)$ است. اگر دو توالی مرتب شده از صفرها و یک ها موجود باشد، می توان با اتصال توالی اول با معکوس توالی دوم، یک توالی دو آهنگی تولید کرد. این توالی تولید شده، با استفاده از مرتب ساز دو آهنگی مرتب خواهند شد.

شبکه های مرتب سازی $\text{SORTER}[n]$ شبکه های مقایسه ای هستند که براساس شبکه های ادغام $\text{MERGER}[n]$ مبتنی بر الگوریتم مرتب ساز ادغامی پیاده سازی و در هر مرحله، به صورت بازگشتی 2^n توالی را مرتب می کند. هر $\text{SORTER}[n]$ از ادغام تولید می شود. شبکه مقایسه ای که میتواند هردنباله صعودی یکنواخت از صفرها و یک ها را ادغام کند، می تواند هر دنباله صعودی یکنواخت از اعداد دلخواه را نیز ادغام کند.

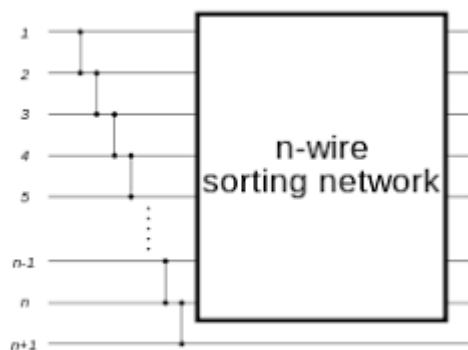


اندازه شبکه های مقایسه ای بوسیله تعداد نهایی مقایسه گر ها و تعداد لایه های شبکه به دست میآید. در هریک از این دو پارامتر، پیدا کردن شبکه مرتب سازی بهینه حتی زمانی که شبکه، به تعداد کمی ورودی محدود می شود، به عنوان یک چالش جدی به شمار میآید.

برخی شبکه های مرتب سازی که از تعداد کم مقایسه گر برخوردار هستند، از الگوریتم های تکاملی برای تولید شبکه ها استفاده میکنند.

اصولا شبکه های مرتب ساز بهینه برای ورودی های کوچک و با تعداد ثابت n با عمق حداقلی و یا اندازه حداقلی میتوانند ایجاد شوند.

این شبکه ها میتوانند برای افزایش کارای در شبکه های مرتب ساز بزرگتر، با ترکیب های بازگشتی، به صورت پایه مورد استفاده واقع شوند. نتیجه می گیریم عمل half clean روی دنباله Bitonic همواره نیمه بالایی از نیمه پایینی کوچکتر است، و همواره یک نیمه مرتب شده است بدین معنی که همه عناصر آن یا صفر است یا یک.



۱- نقش الگوریتم برای تأثیرگذاری مهندسی نرم‌افزار در تحقق شعار سال (تولید، پشتیبانی‌ها و مانع زدایی‌ها)

چه می‌تواند باشد؟

به گزارش مشرق، قریب به ده سال است که مقام معظم رهبری شعارهای ابتدای هر سال را با رویکرد اقتصادی مشخص کرده‌اند. برای مثال جهاد اقتصادی، تولید ملی و حمایت از کار و سرمایه ایرانی، حماسه سیاسی و حماسه اقتصادی، اقتصاد و فرهنگ، با عزم ملی و مدیریت جهادی، اقتصاد مقاومتی؛ اقدام و عمل، تولید اشتغال و... نام‌هایی بودند که توسط مقام معظم رهبری برای ابتدای هر سال انتخاب شدند.

سال ۱۴۰۰ را سال «تولید؛ پشتیبانی‌ها، مانع‌زدایی‌ها» نام‌گذاری و تأکید کردند که «شعار انقلابی جهش تولید» امسال باید با حمایت همه‌جانبه و رفع موانع، کامل محقق شود. انتخاب شعار حکیمانه امسال توسط مقام معظم له نشان می‌دهد که کشور بیش از همیشه نیازمند همدلی، همکاری و تلاش مضاعف است تا با ریل‌گذاری اقتصادی به سمت تولید حرکت کند. در عرصه فرهنگ با زمینه‌سازی برای پشتیبانی از نیروهای خلاق و دارندگان ایده و نیز با توسعه محصولات فرهنگی و خلاق می‌توان به رشد و توسعه کسب‌وکارها و اقتصاد این بخش دست‌یافت. امری که در دو سال اخیر تلاش شد تا ضرورت و الزامات زیرساختی آن در سطح دستگاه و دولت طرح شود، مورد توجه و بررسی قرار گیرد و به‌عنوان معاون وزیر و مسئول بهبود محیط کسب‌وکارهای وزارتخانه نیز بر این فرایندها نظارت دقیق داشته باشند چراکه رویکرد توسعه اقتصاد فرهنگ و توسعه صنایع فرهنگی و خلاق در کشور ما ازجمله انگاره‌های نوین در مباحث رشد شاخص‌های اقتصادی و تولیدی است.

یکی از راه‌کارهای مطرح‌شده که شاید شاهد تحقق تولید، پشتیبانی‌ها، مانع‌زدایی‌ها، در پایان سال ۱۴۰۰ باشیم: با خروج تولید از مظلومیت فعلی و رفع موانع مهمی چون پایین بودن سهم سرمایه‌گذاری نسبت به GDP، واردات غیرضروری، مقابله با قاچاق، اهلیت و ... و در بحث پشتیبانی از تولید توجه به اصولی چون صنعتی شدن و جلوگیری از خام‌فروشی، خصوصی‌سازی به معنای واقعی مبتنی بر حفظ نیروی کار و بازسازی و نوسازی واحد تولیدی و به‌کارگیری تکنولوژی و نوآوری، کارگیری سیستم بانکی در راستای حمایت از تولید و فناوری، راهبرد صادرات با به‌کارگیری تمام توان دیپلماسی و شناسایی و تسخیر بازارهای صادراتی و

سال ۱۴۰۰، سال تولید، پشتیبانی‌ها، مانع‌زدایی‌ها نام‌گذاری شده است و پرواضح است که برای تحقق این شعار بایستی به کلیه عوامل مؤثر در تولید توجه شود و فقط مقصود از این شعار، پشتیبانی از کارفرمایان که عامل سرمایه را در مجموعه عوامل مؤثر بر

تولید نمایندگی می‌کنند، نبوده است و بایستی به کارگران و کارمندان شاغل در بخش‌های تولیدی اعم از تولید محصولات و تولید خدمات نیز توجه و التفات لازم صورت پذیرد. در تبیین عوامل تولید نظریات مختلفی وجود دارد ولیکن در تمام سناریوهای مربوط به تعیین عوامل مؤثر در تولید، دو عامل کار و سرمایه به‌عنوان اصلی‌ترین عوامل مطرح بوده و هستند.

بهترین اقدام دولت برای حمایت از تولید و مانع زدایی از مسیر آن، مدیریت ریسک‌هایی است که به‌واسطه فعالیت‌های دولت به اقتصاد و به‌ویژه بخش خصوصی تحمیل می‌شود.

دو سرفصل کلیدی در ریسک‌های ناشی از فعالیت‌های دولت برای اقتصاد، تغییر ناگهانی مقررات و نقض عهدهای دولت در قراردادهای، به‌ویژه قراردادهای سنگین سرمایه‌گذاری است که مهم‌ترین مصداق آن در اقتصاد امروز ایران واگذاری‌های اموال دولتی به بخش خصوصی محسوب می‌شود.

خوب است دولتمردان و این نکته را در نظر داشته باشند که مقررات و قراردادهای در روز نخست برای پیش‌بینی پذیر کردن فضای کسب‌وکار و به‌عنوان ابزاری برای مدیریت و کاهش ریسک ساخته شده‌اند؛ اما تغییرات ناگهانی و تصمیمات خلق‌الساعه در عرصه مقررات گذاری و نقض عهد در قراردادهای دولتی، به‌ویژه واگذاری‌ها، با هر بهانه و به هر مصلحتی که انجام می‌شوند درواقع نقض غرض اولیه مقررات گذاری و انعقاد قرارداد و موجب افزایش ریسک و کاهش کیفیت فضای کسب‌وکار هستند.

دولتمردان باید این نکته را همواره به خاطر داشته باشند که برای فعالان اقتصادی پاداش تولید همواره در برابر ریسک و مخاطرات آن سنجیده می‌شود و در این موازنه اساسی‌ترین امکان اصیل و اصولی دولت برای حمایت از تولید و مانع زدایی از مسیر آن کاهش ریسک و ایجاد امنیت اقتصادی برای بازیگران این صحنه است.

در سال ۱۴۰۰ پشتیبانی‌ها و مانع زدایی‌ها نیازمند حمایت‌های سخت‌افزاری و نرم‌افزاری خاص است که همه باید به آن عمل کنند و در صورت تحقق شعار سال شاهد تحول در کشور خواهیم بود. این مسئله مانند قانون است، وقتی قانونی در مجلس مشخص می‌شود وظیفه دولت است تا آیین‌نامه‌های اجرایی آن را تعیین نماید. اگر شعار سال ۱۴۰۰ تحقق پیدا کند تحول بسیار خوبی در کشور به وجود خواهد آمد.

در دنیای امروز مسائل فراوانی وجود دارد که می‌توان توسط الگوریتم‌ها راه‌حل‌های مناسب و بهینه‌ای برای آن‌ها ارائه نمود. استفاده از الگوریتم‌ها در این مسائل باعث صرفه‌جویی در وقت و هزینه شده و راهکارهای نوینی را پیش رو قرار می‌دهد. ازجمله:

پزشکی و ژنتیک، تجارت الکترونیک، اینترنت، صنعت، مسیریابی، مرتب‌سازی و ...

نقش الگوریتم در زندگی امروزی و برای رسیدن به هدف بسیار زیاد است، حتی با یک وب گردی ساده هم متوجه اهمیت الگوریتم در زندگی امروزی خواهیم شد؛ به‌طور مثال الگوریتم موفقیت در کنکور، الگوریتم رسیدن به موفقیت، الگوریتم رسیدن به هدف،

الگوریتم موفقیت در پروژه و

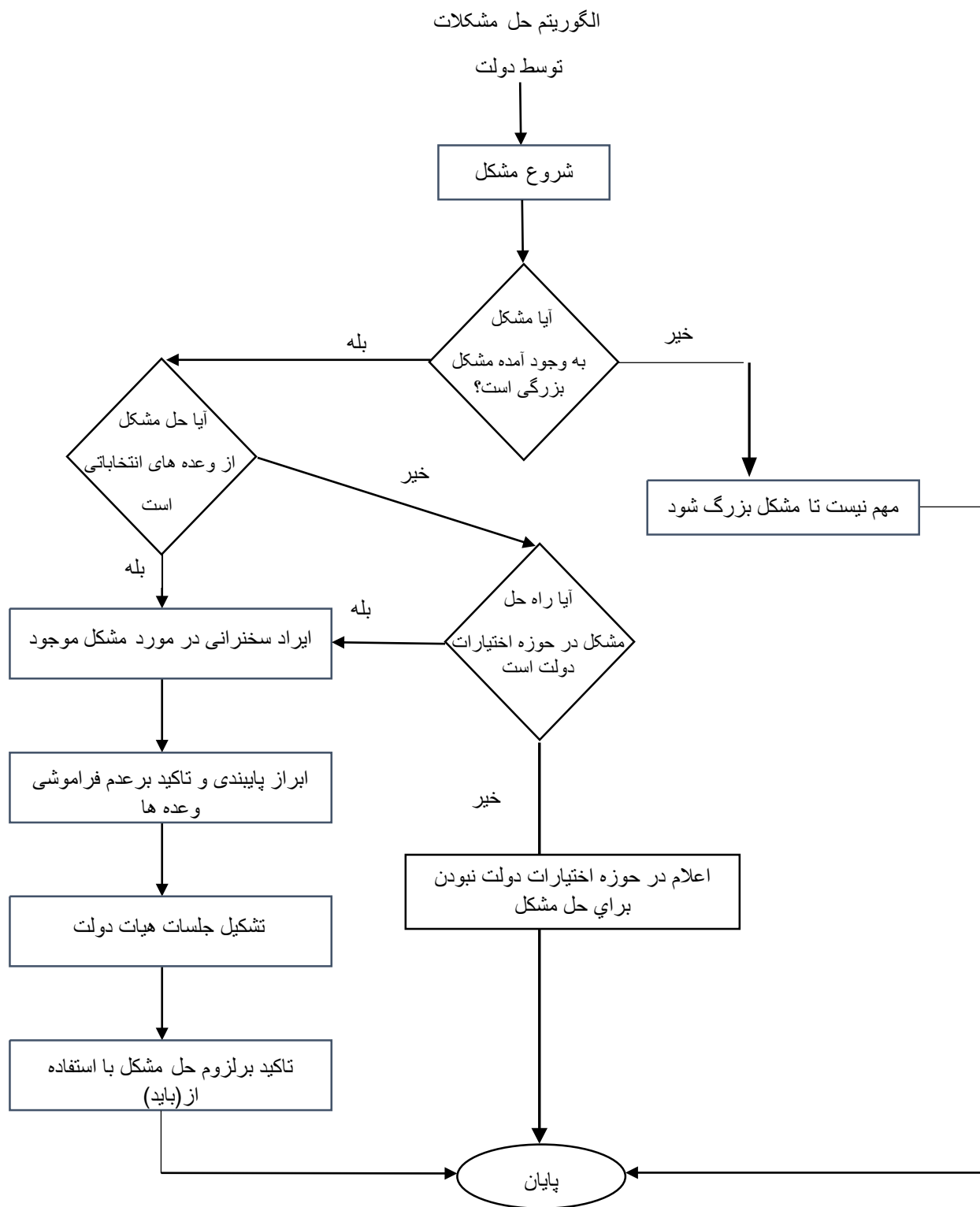
لوک دورمل نویسنده و روزنامه‌نگاری که در حوزه علوم انسانی دیجیتال مشغول به فعالیت است، آخرین کتاب او با عنوان "فرمول: چگونه الگوریتم‌ها تمامی مسائل ما را حل می‌کنند" می‌باشد.

علیرغم تمامی پیشرفت‌های فناورانه و ظهور ماشین‌ها پس از انقلاب صنعتی و ظهور کامپیوترها و ربات‌ها پس از عصر اطلاعاتی و شبکه‌ای و شکل‌گیری فضای مجازی کسب‌وکار در کنار فضای واقعی کسب‌وکار، هنوز دو عامل سرمایه و کار جزو عوامل مؤثر و غالب تحقق تولید مطرح می‌باشند.

کشور اکنون نیازمند بازخوانی قوانین، روزآمدسازی ساختارهای دولتی و رویکردهای نو شونده در اقتصاد خلاق و مدیریت درآمدهای غیرنفتی است. در چنین وضعیتی برداشتن گام‌های نوین برای توسعه تولیدات و راه یافتن به جرگه کشورهای صادرکننده محصولات و خدمات فرهنگی به قدرت بالا و مهارت مدیریتی نیاز دارد.

باوجود یک راهکار و الگوریتم مناسب برای ارائه راه‌حل و برطرف نمودن مشکلات، تا حد زیادی می‌توان از موانع عبور کرده و به اهداف و در نتیجه تحقق شعار سال نزدیک‌تر شد.

برای نمونه یک الگوریتم در ادامه نشان داده شده است.



۲- با ذکر مثال و ارائه یک الگوریتم اولیه (لزومی به اثبات درستی الگوریتم ارائه شده وجود ندارد) تلاش نمایید قدمی برای حل مشکل کارآمدی دانشجویان مهندسی نرم افزار برای مشارکت در تولید نرم افزار در دوران دانشجویی و یا پس از آن ارائه نمایید؟ (استفاده از الگوریتم های مطالعه شده در درس مورد استقبال خواهد بود)

در طی ۱۰ سال گذشته نقش کلیدی که کاربرد فناوری اطلاعات و ارتباطات می تواند در توسعه اجتماعی و اقتصادی کشورها ایفاء نماید، مشخص و قطعی شده است. این امر باعث شده تا میزان توسعه یافتگی کشورها از اهمیت برخوردار شده و مدیران هر کشور و طرف های تجاری و سرمایه گذاران در کشورها مایل باشند تا به طور شفاف از وضعیت کاربرد فناوری اطلاعات و ارتباطات مطلع شوند. در پاسخ گویی به این نیاز از یک طرف سازمان ها و مؤسسات تحقیقاتی روش ها و شاخص های مختلفی را برای تعیین میزان پیشرفته بودن کاربرد فناوری اطلاعات تعریف نموده با اتکال به آن کشورهای مختلف دنیا را طبقه بندی نمایند. از طرف دیگر کشورها خود نیز با جدوجهد در تلاش بوده اند تا وضعیت «رقومی» خود را هرچه بیش تر شفاف نمایند.

اگرچه در ایران از دیرباز مرکز آمار ایران مسئولیت تهیه آمارهای مورد نیاز کشور را بر عهده داشته و در همین ارتباط سرشماری ها و مطالعات مختلفی را انجام داده و هر ساله کتاب آمار ایران را چاپ نموده است، لیکن جدید بودن مباحث مرتبط با ICT و تغییرات سریع در حوزه آن باعث شده تا نظام موجود آماری کشور قادر به پاسخ گویی به نیازهای اطلاعاتی موجود نباشد. برای حل این مشکل و به روز نگه داری دائمی آمارهای مورد نیاز برنامه ریزان کشور، چند سالی است که دولت ایجاد یک نظام جامع اطلاعاتی (اعم از آماری و غیر آماری) را مدنظر قرار داده است. در همین ارتباط مطالعات اولیه انجام شده و بر اساس آن به زودی مرکزی به نام مرکز ملی اطلاعات ایجاد خواهد شد.

در طی سال های گذشته بخش خدمات و نرم افزار در ایران در مقایسه با سخت افزار از رشد کمتری برخوردار بوده، لیکن در طی دو سال گذشته این روند تغییر کرده و هم اکنون نرم افزار با رشدی معادل ۴۱٪ در سال به نسبت از سایر بخش ها رشد بیش تری دارد. در این بخش ده ها شرکت به تولید و عرضه نرم افزار، محتوا و خدمات اشتغال دارند. تعداد شرکت های نرم افزاری فعال در سطح کشور بالغ بر ۳۰۰ شرکت است. اغلب شرکت ها در این بخش خود اقدام به فروش مستقیم نرم افزارهای تولیدی خود می نمایند. تعداد شرکت های خارجی نرم افزاری فعال در بازار نرم افزار ایران کم است و آن تعداد کم هم بیش تر در زمینه های نظیر بانکداری، و بنگاه های اقتصادی بزرگ فعالیت دارند. دولت برنامه های حمایتی مختلفی را برای صنعت نرم افزار تدارک دیده است. از جمله اعمال جدی قانون منع تکثیر غیر مجاز نرم افزارها، حمایت از تولید محصولات جدید و یا ارتقای محصولات موجود در بازار، وجود معافیت های مالیاتی برای تولید کنندگان نرم افزار و محتوا، و اعطای وام های کم بهره را می توان از جمله اقدامات دولت در این خصوص دانست. علاوه بر این دولت با اعلام طرح " (initiative) تکفا" که برای توسعه کاربرد فناوری اطلاعات در دولت و جامعه تهیه و تدوین شده

است، بازار نرم‌افزار را رشد زیادی داده است. مهم‌ترین انواع خدمات ارائه‌شده را می‌توان موارد زیر دانست:

مشاوره، مهندسی مجدد فرآیندها، تحلیل، طراحی و پیاده‌سازی سیستم، تولید نرم‌افزار بر طبق سفارش، نظارت بر اجرای پروژه‌ها و آموزش.

فعالیت مشاوره و نظارت بر اجرای پروژه‌ها توسط تعداد نسبتاً اندکی از شرکت‌ها و نیز کارشناسان مستقل صورت می‌گیرد. این گروه از شرکت‌ها عمدتاً خود را در دسته "شرکت‌های مشاور نرم‌افزاری" طبقه‌بندی می‌نمایند. فعالیت مهندسی مجدد فرآیندها عمدتاً توسط آن گروه از شرکت‌ها صورت می‌گیرد که بازار خود را بیش‌تر صنایع می‌دانند. تحلیل، طراحی و پیاده‌سازی سیستم توسط اغلب شرکت‌های تولیدکننده نرم‌افزار صورت می‌گیرد. خدمات آموزشی از جمله مهم‌ترین انواع خدمات ارائه‌شده در ایران است. بر اساس بررسی‌های به‌عمل‌آمده در حدود ۱۰۰ مؤسسه آموزش فناوری اطلاعات در کشور مشغول فعالیت هستند. حجم کلی بازار نرم‌افزار و خدمات ایران در حال حاضر حدود ۳۰۰ میلیون دلار در سال تخمین زده می‌شود.

از جمله حرکت‌های مهمی که در زمینه نرم‌افزار و خدمات در کشور آغاز شده، حرکت و تلاش جمعی شرکت‌های نرم‌افزاری برای دستیابی به صادرات کالاها و خدمات نرم‌افزاری است. آنان برای تحقق این مهم شرکت «تحقیقات و توسعه صادرات نرم‌افزار ثنارای» را ایجاد نمودند. اقدامات شرکت ثنارای با پشتیبانی جدی دولت منجر به حضور شرکت‌های ایرانی در نمایشگاه‌های بین‌المللی در حوزه فناوری اطلاعات و بازاریابی برون‌مرزی کالاها و خدمات نرم‌افزاری ایران شده است.

قدمت انفورماتیک و مقررات مربوط به آن به بیش از سی سال پیش بازمی‌گردد. اما جدی‌ترین قوانین این حوزه در طی ۲۵ سال اخیر و پس از وقوع انقلاب اسلامی در کشور به تصویب رسیده‌اند.

مشکلات تولید نرم‌افزار: افزایش سطح آگاهی متخصصان عرصه فناوری اطلاعات با تکنولوژی روز، استفاده از قانون‌های منسجم و یکدست، رعایت قانون کپی‌رایت و در نظر گرفتن یک قانون متمرکز ناظر بر کلیه نهادهای مختلف در برنامه‌ریزی و مدیریت، هم‌چنین قانون‌گذاری در زمینه IT راهکارهایی مناسب برای کاهش مشکلات IT در کشور به حساب می‌آیند. تولید نرم‌افزار یا به‌طور محدودتر متدولوژی تولید نرم‌افزار، منحنی شناخته‌شده برای متخصصین و دست‌اندرکاران است. در ایران بسیاری از افراد در این زمینه اطلاعات کافی دارند و در جریان آخرین دستاوردها و مقالات روز هستند، اما در پیاده‌سازی سیستم‌های در داخل ایران، ردپای این صحبت‌ها کمتر دیده می‌شود. به گفته کارشناسان، صنعت نرم‌افزار در ایران با مشکلات زیرساختی عدیده‌ای روبروست، اما تلاش‌های بسیاری برای رفع این مشکلات و رسیدن به ایده آل‌ها صورت گرفته است. از مشکلات پیش روی شرکت‌های تولیدکننده نرم‌افزار در ایران می‌توان مشکلاتی همچون بالا بودن هزینه‌های تولید نرم‌افزار، به طول انجامیدن مدت‌زمان تولید، عدم حمایت‌های دولتی، عدم ارتباط با تولیدکنندگان خارجی، تولیدات سفارشی، خدمات و پشتیبانی پس از فروش، عدم رعایت

قانون کپی‌رایت، کمبود نیروهای متخصص، نبود پشتیبانی‌های لازم را نام برد که باعث شده صنعت نرم‌افزار ایران نتواند به جایگاه واقعی خود دست پیدا کند.

چالش‌هایی که برخی از آن‌ها به سیاست‌های کلان کشور مرتبط و برخی دیگر زائیده ویژگی‌های خاص نرم‌افزار و دست‌اندرکاران تولید و توسعه آن است.

یکی از کارشناسان در این ارتباط می‌گوید: تولید نرم‌افزار به دلیل عدم استفاده از ابزار و انرژی فیزیکی قابل‌لمس نیست و از این رو، تولید آن از نگاه عام سهل به نظر می‌رسد اما در واقعیت تولید یک نرم‌افزار که حجمی کمتر از یک CD را به خود اختصاص می‌دهد، بعضاً ماه‌ها کار یا چند سال کار کارشناسان را به خود اختصاص داده است.

شکست در پروژه‌های نرم‌افزاری در هر یک از چهار مورد «هزینه»، «زمان»، «کیفیت» و «دستیابی به اهداف» مطرح می‌شود؛ بدین معنا که اگر پروژه‌های با صرف هزینه بیشتر یا زمان بیشتر یا با کیفیت پایین‌تر انجام شود، علیرغم به پایان رسیدن پروژه، آن را توأم با شکست می‌دانیم.

به‌طور کلی دلایل اصلی شکست پروژه‌های IT در ایران را می‌توان به دودسته عوامل داخلی و خارجی تقسیم کرد:

عوامل داخلی: مدیران پروژه کم‌تجربه، ناتوانی‌های شرکت‌های تولید نرم‌افزار، قراردادهای ناپخته، کمبود نیروی انسانی متخصص، مقاومت‌های کاربران و سازمان‌ها در پذیرش سیستم‌های جدید، ارتباط با مشتریان و کاربران و عدم درگیر کردن کاربران در پروژه می‌باشد.

عوامل خارجی: نبود سرمایه‌گذاری مناسب برای پژوهش و تحقیق در حوزه نرم‌افزار، سرمایه‌گذاری کم در بخش خصوصی و عدم حمایت دولت، عدم استفاده از یک استاندارد واحد، مشکلات حضور در مناقصات بین‌المللی، ارزان بودن نرم‌افزار و عدم در نظر گرفتن آن به‌صورت یک کالا، تعطیلات رسمی و برنامه‌ریزی زمانی، عدم شناسایی حقوق مولفین و قانون کپی‌رایت، فیلترینگ و تحریم ایران و مشکلات موجود کشور در زمینه مستندسازی تولید محصولات نرم و رعایت نکردن مستندات تعریف شده نرم‌افزاری می‌باشد.

فرآیندهای مدیریت پروژه با ۹ توانمندی تعریف می‌شود که این توانمندی‌ها عبارتند از: مدیریت یکپارچگی پروژه، محدوده، زمان، هزینه، کیفیت، منابع انسانی، ارتباطات، ریسک و برون‌سپاری و از آنجایی که نقش هر یک از عوامل فوق در تولید یک محصول نرم‌افزاری کلیدی است توانمندی یک مدیر پروژه در تولید نرم‌افزار یکی از عوامل مهم و حیاتی در موفقیت پروژه به شمار می‌رود. اکثر مدیران پروژه نرم‌افزاری در ایران، برنامه‌نویسان قدیمی‌تر و یا قوی‌تر گروه هستند و همین یک نفر در تولید یک نرم‌افزار نقش‌های زیادی از جمله تحلیل، طراحی و برنامه‌نویسی و ... را بر عهده دارد در حالیکه فرآیند مدیریت دانش و توانایی خاص خود

را می‌خواهد و توانمندی‌های خاص خود را می‌طلبد.

ناتوانی‌های شرکت‌های تولید نرم‌افزار: دست‌اندرکاران تولید نرم‌افزار در کشور ما بیشتر شرکت‌های کوچک نرم‌افزاری با پشتوانه‌های مالی اندک هستند. این شرکت‌ها عمدتاً به دلیل محدودیت منابع از رویکرد توسعه تکنولوژی به‌منظور آشنایی با ابزار و روش‌های نوین تولید نرم‌افزار غافل می‌شوند.

قراردادهای ناپخته: بیشترین مشتریان مهم حوزه نرم‌افزار در کشور ما را سازمان‌های دولتی و نیمه‌دولتی تشکیل می‌دهند. قراردادهای تولید نرم‌افزار، قراردادهایی ناپخته و یک‌طرفه هستند و مجریان چنین پروژه‌هایی، باوجود آگاهی نسبت به ضعف‌های آن بیشتر اوقات به دلایل اقتصادی مجبور به پذیرش آن‌ها هستند.

تجربه‌های ناکافی تیم پروژه: چالش دیگری نیز پیش روی مدیران شرکت‌های کوچک و متوسط وجود دارد، برای تمامی مدیرعاملان مهم است که کارمندان تیزهوش، خلاق و مستعد استخدام کنند. شاید یکی از مهم‌ترین عوامل موفقیت پروژه‌های نرم‌افزاری را افراد نام‌برد. متأسفانه در کشور ما کمبود نیروی متخصص یکی از چالش‌های عمده است. عمدتاً می‌توان عدم به‌روز بودن و دسترسی به تکنولوژی روز و اینکه نیروی کار روی تکنولوژی قدیمی با برنامه‌های آموزشی نامناسب و ناکافی پیوند خورده است را یکی از عوامل ضعف تیم نرم‌افزاری دانست.

متأسفانه ضعف نظام آموزشی در دانشگاه‌ها و اینکه محتوی آموزشی اکثر دروس دانشگاهی کاربردی و عملی نبوده و صرفاً تئوری است و در بازار کار کاربردی ندارد را می‌توان یکی از عوامل ضعف متخصصین ایرانی نام برد.

مقاومت‌های کاربران و سازمان‌ها در پذیرش سیستم‌های جدید: بخش‌های IT همچنان در اغلب سازمان‌ها و مراکز صنعتی، خدماتی و بازرگانی سازمان ضعیفی دارند و از دانش و فناوری روز، در عرصه‌هایی که تکامل و تحول در آن، ساعت‌به‌ساعت رخ می‌دهد، به دور هستند چون پروژه‌های IT به‌منظور برآورده ساختن اهداف کلان سازمان آغاز می‌شوند، نهایتاً باید در سطوح عملیاتی پاسخگوی نیازهای کاربران باشند.

عدم سرمایه‌گذاری مناسب برای پژوهش و تحقیق در حوزه نرم‌افزار: اگرچه مساعدت‌هایی در اختیار بخش‌های مختلف خصوصی و دولتی در چارچوب طرح‌های مختلف قرار گرفته است، عمده این سرمایه‌ها اغلب به سمت بخش‌های سخت‌افزاری سرازیر و در عمل سرمایه‌گذاری در عرصه تولید و به‌تناسب آن تحقیق و پژوهش کاربردی حوزه نرم‌افزار که در زمره سودآورترین عرصه‌های اقتصادی در دنیا شناخته می‌شود، بسیار اندک است.

کشور ما هنوز دوران اولیه بلوغ خود را در عرصه IT تجربه می‌کند، دوره‌ای سرشار از مسائل و چالش‌های گوناگون. چالش‌هایی که برخی از آن‌ها به سیاست‌های کلان کشور مرتبط و برخی دیگر زاینده ویژگی‌های خاص نرم‌افزار و دست‌اندرکاران تولید و توسعه

آن است.

در کشور ما حمایت کافی از بخش خصوصی در زمینه IT به عمل نمی‌آید. برخی مشکلات که این گروه با آن روبه‌رو هستند عبارتند از:

روند طولانی پیشرفت کار در ادارات دولتی، عدم استفاده از یک استاندارد واحد، یکی از ضعف‌های شرکت‌های تولیدکننده نرم‌افزاری این است که تولیدات آن‌ها از استانداردهای لازم برخوردار نیست.

شرکت‌های نرم‌افزاری و مشکلات حضور در مناقصات بین‌المللی: در حال حاضر مشکلات اجرایی بسیاری بر سر راه حضور شرکت‌های نرم‌افزاری جهت حضور در مناقصات بین‌المللی وجود دارد که بخشی از آن به دولت و بخش دیگری از آن به خود شرکت‌های نرم‌افزاری برمی‌گردد. این مشکلات عبارتند از نبود امکان صدور آسان ضمانت‌نامه‌های مختلف ارزی نظیر شرکت در مناقصه، پیش‌پرداخت و حسن انجام کار برای شرکت‌های نرم‌افزاری، عدم حمایت مناسب بانک‌ها، مشکلات شرکت‌های نرم‌افزاری، عدم شناسایی حقوق مولفین و قانون کپی‌رایت.

استانداردها: استاندارد کردن و استاندارد از پایه‌های علم و فناوری است که در پیشرفت صنعت و اقتصاد نقشی به سزا دارد و باید در جهت افزایش سطح کیفیت‌ها تلاش کرد تا به فناوری پیشرفته تولید هر محصول نائل آمد.

یکی از مشکلات شرکت‌های نرم‌افزار ایرانی در بازارهای جهانی عدم آشنایی با استاندارد نرم‌افزار است، متأسفانه استاندارد نرم‌افزار هنوز در ایران به‌درستی تدوین و یا تعریف نشده است و تاکنون هیچ اقدام منسجمی در کشور ما برای تعیین استاندارد تولید نرم‌افزار صورت نگرفته و هیچ اجباری برای ارائه شناسنامه نرم‌افزارها وجود نداشته است. از آنجاکه تنوع حوزه عملکرد در زمینه نرم‌افزار بسیار زیاد است، ادامه حیات این صنعت نمی‌تواند بی‌نیاز از وجود استانداردهای منسجم و یکدست باشد. با توجه به اینکه هر تولیدکننده نرم‌افزاری یک روش تولید خاص خود اتخاذ می‌کند و در مراحل ساخت از های متفاوت استفاده می‌کند، نتیجه کار شرکت‌های مختلف به لحاظ روند کیفی و اجرایی شکل و شمایلی گوناگون خواهد داشت و این امر اصول یکپارچگی در این بازار را فدای خود خواهد کرد.

اینجاست که لزوم وجود استاندارد نرم‌افزار به‌وضوح حس می‌شود. کارشناسان نرم‌افزار معتقدند که پشتیبانی قدرتمند محصولات و همچنین مستندسازی و تخمین کیفیت تنها با وجود یک استاندارد منظم مقدور است، وقتی استاندارد صحیح وجود داشته باشد کار گروهی بیشتر معنا پیدا می‌کند. هم‌اکنون استانداردهای جهانی زیادی مانند IEEE وجود کیفیت و پشتیبانی مناسب در صنعت نرم‌افزار دنیا را تضمین می‌کنند.

متأسفانه به علت فقدان استاندارد برای مستندسازی محصولات نرم‌افزاری و بی‌توجهی سازندگان نرم‌افزار و کارفرمایان،

نرم افزارهای تولیدشده در ایران اکثراً فاقد حداقل مستندات لازم هستند.

یکی از ضعف های شرکت های تولیدکننده نرم افزاری این است که تولیدات آن ها از استانداردهای لازم برخوردار نیست. در ایران شرکت های نرم افزاری اغلب در رده شرکت های کوچک و با تعداد محدودی نیروی فنی شکل می گیرد و سپس پروژه های در دست را تبدیل به یک محصول کرده و به صورت عمومی اقدام به فروش آن می کنند. که این کار اغلب بدون نیازسنجی در بازار صورت می گیرد، بنابراین محصول تولیدشده از استانداردهای لازم برخوردار نیستند.

برای توسعه صنعت نرم افزار باید بر روی فرایند انرژی زیادی گذاشته شود. تولید نرم افزار بزرگ یک کار گروهی است، یک صنعت است اجرای این کار بدون فرایند ممکن نیست بر اساس آمارهای موجود عوامل ریسک در یک پروژه نرم افزاری بزرگ و سهم هر یک از آن ها عبارتند از نیروی انسانی، فرایند کار و فناوری و ابزار روز می باشد.

قسمت عمده از عوامل ریسک در نظر گرفته شده در عامل نیروی انسانی ما فرایند کار قابل کنترل است. بنابراین دیده می شود که نقش فرایند تولید نرم افزار در موفقیت یک پروژه بزرگ نرم افزاری تا چه حد تعیین کننده است.

نکته پایانی که از این اطلاعات استنباط می شود این است که عامل فناوری و ابزار، درصد به مراتب پایین تری را به خود اختصاص داده است. بنابراین در آموزش ها و برنامه ریزی های کار باید توجه بیشتری به فرایند کار کرد تا به ابزار پیاده سازی؛ متأسفانه این امر در بسیاری از اوقات فراموش می شود.

رعایت توصیه های زیر می تواند احتمال شکست پرو ه های IT در کشور را کاهش دهد.

✓ افزایش میزان سطح آگاهی جامعه در مقوله ی فناوری اطلاعات آگاهی

✓ افزایش میزان سطح آگاهی متخصصان عرصه ی فناوری اطلاعات با تکنولوژی روز با تا سیس مؤسساتی تحت حمایت

دولت برای تربیت و آماده سازی نیروی متخصص

✓ استفاده از استانداردهای منسجم و یکدست

✓ سعی هر چه بیش تر در جهت رعایت قانون کپی رایت

✓ در نظر گرفتن یک نهاد متمرکز ناظر بر کلیه ی نهاد های مختلف در برنامه ریزی و مدیریت و قانون گذاری در زمینه ی IT

مهندسی نرم افزار عبارت است از کاربرد مهندسی برای طراحی نرم افزار، توسعه، پیاده سازی و نگهداری از نرم افزار در یک روش سیستماتیک.

استفاده از الگوریتم دایجسترا: روند الگوریتم دایجسترا مطابق زیر می باشد :

انتخاب رأس مبدأ

مجموعه S ، شامل رئوس گراف، معین می‌شود. در شروع، این مجموعه تهی بوده و با پیشرفت الگوریتم، این مجموعه رئوسی که کوتاه‌ترین مسیر به آن‌ها یافت شده است را در برمی‌گیرد. رأس مبدأ با اندیس صفر را در داخل S قرار می‌دهد. برای رئوس خارج از S ، اندیسی معادل، طول یال + اندیس رأس قبلی، در نظر می‌گیرد. اگر رأس خارج از مجموعه دارای اندیس باشد، اندیس جدید کمترین مقدار از بین اندیس قبلی و طول یال + اندیس رأس قبل، می‌باشد. از رئوس خارج مجموعه، رأسی با کمترین اندیس انتخاب‌شده و به مجموعه S اضافه می‌گردد. این کار را دوباره از مرحله ۴ ادامه داده تا رأس مقصد وارد مجموعه S شود. در پایان اگر رأس مقصد دارای اندیس باشد، اندیس آن نشان‌دهنده‌ی مسافت بین مبدأ و مقصد می‌باشد. در غیر این صورت هیچ مسیری بین مبدأ و مقصد موجود نمی‌باشد.

همچنین برای پیدا کردن مسیر می‌توان اندیس دیگری برای هر رأس در نظر گرفت که نشان‌دهنده‌ی رأس قبلی در مسیر طی شده باشد. بدین ترتیب پس از پایان اجرای الگوریتم، با دنبال کردن رئوس قبلی از مقصد به مبدأ، کوتاه‌ترین مسیر بین دونقطه نیز یافت می‌شود.

۳- دسته‌های سختی مسائل و تفاوت آن‌ها را تشریح نمایید.

به‌طور کلی مسائل به دودسته حل‌شدنی و حل‌ناشدنی تقسیم می‌شوند. مسائل حل‌شدنی هم به دو قسمت رام‌شدنی و رام‌ناشدنی یا بغرنج تقسیم می‌شوند. برای مسائل حل‌ناشدنی هیچ الگوریتمی راه‌حلی وجود ندارد. مسائل رام‌شدنی هم در تئوری و هم در عمل حل‌شدنی‌اند. مسائل رام‌نشده یا بغرنج در تئوری حل‌شدنی‌اند، اما به دلیل حجم عظیم زمان موردنیاز، در عمل حل‌شدنی نیستند.

پیچیدگی زمانی یک مسئله تعداد گام‌های موردنیاز برای حل یک نمونه از یک مسئله به‌عنوان تابعی از اندازه‌ی ورودی (معمولاً به‌وسیله تعداد بیت‌ها بیان می‌شود) به‌وسیله کارآمدترین الگوریتم می‌باشد. برای درک بهتر این مسئله، فرض کنید که یک مسئله با ورودی n بیت در n^2 گام حل شود. در این مثال می‌گوییم که مسئله از درجه پیچیدگی n^2 می‌باشد. البته تعداد دقیق گام‌ها بستگی به ماشین و زبان مورد استفاده دارد. اما برای صرف‌نظر کردن از این مشکل، نشانه‌گذاری O بزرگ (Big O notation)

معمولاً بکار می‌رود. اگر یک مسئله پیچیدگی زمانی از مرتبه $O(n^2)$ روی یک کامپیوتر نمونه داشته باشد، معمولاً روی اکثر کامپیوترهای دیگر نیز پیچیدگی زمانی از مرتبه $O(n^2)$ خواهد داشت.

بحث پیچیدگی زمانی مسائل کلاً یک بحث مربوط به مسائل تصمیم‌گیری (Decision Problems) است نه مسائل بهینه‌سازی (Optimization Problems)

اکثر مسائلی که با آن‌ها برخورد می‌کنیم از نوع بهینه‌سازی‌اند. یعنی می‌خواهیم یک یا چند پارامتر را حداکثر یا حداقل سازیم. زمان تابع چندجمله‌ای ساده‌ای (n^k) از اندازه ورودی (n) است. الگوریتم زمانی چندجمله‌ای، الگوریتمی است که در بدترین حالت، پیچیدگی زمانی آن تابع چندجمله‌ای از اندازه ورودی باشد. مثال‌های الگوریتم زمان - چندجمله‌ای:

$O(n^2)$, $O(n^3)$, $O(1)$, $O(n \log n)$

مثال‌های الگوریتم‌های زمان - غیر چندجمله‌ای:

$O(2^n)$, $O(n^n)$, $O(n!)$

مسائل رام شدنی یا tractable را می‌توان با الگوریتم زمان چندجمله‌ای حل کرد. الگوریتم زمانی چندجمله‌ای، الگوریتمی است که در بدترین حالت پیچیدگی زمانی آن تابع چندجمله‌ای از اندازه ورودی باشد. مسئله‌ای بغرنج است که حل آن توسط الگوریتمی بازمان چندجمله‌ای غیرممکن باشد. ممکن است برای حل مسئله الگوریتم‌های متعددی با زمان‌های مختلف وجود داشته باشد؛ مثلاً حل مسئله‌ی سری فیبوناچی با روش تقسیم و حل از مرتبه‌ی نمایی و با روش پویا از مرتبه خطی است، پس مسئله فیبوناچی از نوع بغرنج نیست. برای آنکه مسئله‌ای بغرنج باشد باید اثبات کرد که هیچ الگوریتمی با مرتبه چندجمله‌ای برای آن وجود ندارد. از نظر بغرنج بودن یا نبودن، مسائل علوم کامپیوتر سه دسته هستند:

۱. مسائلی که الگوریتم‌هایی با پیچیدگی زمانی چندجمله‌ای برای آن‌ها پیدا شده است.
۲. مسائلی که اثبات شده است که بغرنج هستند، یعنی الگوریتم چندجمله‌ای برای آن یافت نمی‌شود.
۳. مسائلی که بغرنج بودن آن‌ها ثابت نشده است ولی از طرف دیگر هیچ الگوریتم چندجمله‌ای هم برای آن‌ها پیدا نشده است. مثلاً جست‌وجوی دودویی در آرایه مرتب با مرتبه زمانی $\log n$ ، مسئله مرتب‌سازی آرایه‌ها با مرتبه زمانی $n \log n$ ، ضرب زنجیره‌ای ماتریس‌ها با مرتبه زمانی n^3 دسته اول هستند. همچنین مسائل کوتاه‌ترین مسیر فلویید با مرتبه زمانی n^3 ، درخت دودویی بهینه n^3 و مسئله درخت پوشای کمینه پریم n^2 باینکه الگوریتم‌های نمایی نیز دارند ولی چون برای آن‌ها الگوریتم‌هایی از نوع چندجمله‌ای یافت شده است از این دسته‌اند.

تعداد مسائلی که جزو دسته دوم بوده و بغرنج بودن آنها اثبات شده است، نسبتاً کم بوده است.

مسئله کوله‌پشتی صفر و یک و مسئله فروشنده دوره‌گرد از دسته سوم هستند.

چند نکته:

- همه مسائل P را رام شدنی می‌گویند. درواقع رام شدنی و قابل حل در زمان چندجمله‌ای مترادف هم هستند.

- مسائلی که P نیستند یعنی راه حل مرتبه زمانی چندجمله‌ای ندارند، یا رام نشدنی و بغرنج‌اند (Intractable) یا کلاً حل ناپذیرند

(Unsolvable). مسائل غیر P فقط برای ورودی‌های کوچک به صورت منطقی قابل حل‌اند و یا ابداً قابل حل نیستند.

مسائل رام نشدنی (Intractable) را بر اساس درجه سختی آنها به دسته‌های مختلف تقسیم می‌کنند:

▪ NP

▪ NP-Complete یا NP کامل

▪ NP-hard یا سخت

مسائل P کلاسی از مسائل تصمیم‌گیری هستند که هرکدام می‌تواند با یک ماشین تورینگ قطعی در زمان چندجمله‌ای حل شود.

مسائل NP کلاسی از مسائل تصمیم‌گیری هستند که هرکدام از آنها می‌تواند توسط یک ماشین تورینگ غیرقطعی در زمان چندجمله‌ای حل شود.

مسائل NP کلاسی از مسائل تصمیم‌گیری هستند که هرکدام از این مسائل می‌تواند با یک الگوریتم غیرقطعی زمان چندجمله‌ای حل شود.

معروف‌ترین کلاس‌های پیچیدگی، P و NP هستند که مساله‌ها را از نظر زمان مورد نیاز تقسیم‌بندی می‌کنند. به طور شهودی می‌توان گفت P کلاس مساله‌هایی است که الگوریتم‌های سریع برای پیدا کردن جواب آنها وجود دارد. اما NP شامل آن دسته از مساله‌هاست که اگرچه ممکن است پیدا کردن جواب برای آنها نیاز به زمان زیادی داشته باشد اما چک کردن درستی جواب به وسیله یک الگوریتم سریع ممکن است. البته کلاس‌های پیچیدگی به مرتبه سخت‌تری از NP نیز وجود دارند.

منظور از کلاس P تمام مسائل تصمیم‌گیری‌اند که الگوریتمی با پیچیدگی زمانی چندجمله‌ای برای آنها موجود باشد. با این تعریف طبیعی است که عباراتی مانند n^n یا $n!$ یا 2^n جزو چندجمله‌ای‌ها محسوب نمی‌شوند.

به طور کلی مسائلی که عضو این کلاس هستند توسط یک ماشین تورینگ قطعی پذیرفته می‌شوند و بنابراین ما الگوریتمی در زمان چندجمله‌ای برای این مسائل داریم .

کلاس **Np** (مخفف Nondeterministic Polynomial) : به آن دسته از مسائل تصمیم‌گیری می‌گویند که با داشتن یک جواب این توانایی را داریم که درستی یا نادرستی جواب را با استفاده از یک الگوریتم با پیچیدگی زمانی چندجمله‌ای تعیین کنیم.

تعریف NP	
یک مجموعه $L \in NP$ است اگر یک الگوریتم A و یک چندجمله‌ای $p()$ وجود داشته باشند	
برای همه $x \in L$	برای همه $x' \notin L$
یک y وجود داشته باشد با شرط $ y \leq p(x)$	یک y' وجود داشته باشد با شرط $ y' \leq p(x')$
طوری که $A(x,y) = YES$ در زمان $p(x)$	طوری که $A(x',y') = NO$ در زمان $p(x')$

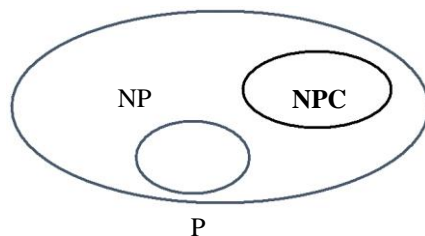
NP مجموعه‌ی مسائل تصمیم‌گیری است که در زمان چندجمله‌ای و با استفاده از ماشین غیرقطعی تورینگ قابل حل شدن هستند. کلاس پیچیدگی P یکی از اعضای NP است اما NP شامل کلاس‌های مهم دیگری نیز هست. که پیچیده‌ترین آن‌ها NP-Complete است بطوریکه برای آن‌ها هیچ الگوریتم شناخته‌شده قابل اجرا در زمان چندجمله‌ای وجود ندارد.

یک الگوریتم غیرقطعی شامل دو فاز است:

- فاز یک: حدس زدن
- فاز دو: چک کردن

اگر در مرحله چک کردن یک الگوریتم غیرقطعی، پیچیدگی زمانی چندجمله‌ای بود، پس به این الگوریتم NP می‌گوییم یعنی چندجمله‌ای غیرقطعی.

در بیان یک الگوریتم غیرقطعی ممکن است از همه دستورها و به‌ویژه از دستورهای choice، failure و success استفاده نشود؛ بنابراین هر الگوریتم معین توسط یک کامپیوتر غیرقطعی قابل اجرا است. پس P زیرمجموعه NP است. محققین زیادی سعی کرده‌اند ثابت کنند $P = NP$ اگر این مسئله ثابت شود مفهوم آن این است که هر مسئله‌ای که برای آن الگوریتم غیرقطعی با مرتبه زمانی چندجمله‌ای وجود دارد می‌توان برای آن یک الگوریتم معین با مرتبه زمانی چندجمله‌ای نیز پیدا کرد.



رابطه بین کلاس‌های P , np , npc

از مهم‌ترین مسائل کلاس NP می‌توان به مسئله فروشنده دوره‌گرد (Salesman Problem)، تجزیه اعداد صحیح (Integer Factorization) و هم‌ریختی گراف‌ها (Graph Isomorphism) اشاره کرد.

مسئله P خودش با پیچیدگی زمانی چندجمله‌ای قابل حل می‌باشد، پس در صورت داشتن جواب می‌توان درستی یا نادرستی آن را تعیین کرد. بنابراین کلاس P حتماً زیرمجموعه کلاس NP می‌باشد. این سؤال را می‌شود به این شکل مطرح کرد که آیا برای هر سؤال NP، می‌توان الگوریتمی از پیچیدگی زمانی چندجمله‌ای پیدا کرد؟

اگر بتوانیم جواب این سؤال را پیدا کنیم در نتیجه می‌توانیم بگوییم که آیا $P=NP$ هست یا خیر.

این سؤال که آیا مسائل کلاس P دقیقاً همان مسائل کلاس NP می‌باشند، یکی از مهم‌ترین مسائل بدون جواب علوم کامپیوتر می‌باشد.

مسائل کلاس NP کامل یا NPC: آن دسته از مسائلی هستند که در صورت پیدا شدن الگوریتم از درجه چندجمله‌ای، می‌توان تمام مسائل NP را به این مسئله تبدیل کرد و با پیچیدگی چندجمله‌ای حل کرد.

به روشی که مسائل NP به NP-hard تبدیل می‌شوند Reduction می‌گویند. به عبارتی فرض کنید که یک مسئله داریم با یک سری ورودی خاص مثلاً X و یک مسئله دیگر داریم با ورودی Y. حال اگر بتوانیم X را به شکلی با درجه چندجمله‌ای، به Y تبدیل کنیم می‌گوییم تکنیک Reduction را به کار برده‌ایم.

به آن الگوریتم از درجه چندجمله‌ای $O(n^c)$ (که c یک ثابت است). که برای Reduction (یا تبدیل ورودی ۲ مسئله به هم) استفاده می‌کنیم، الگوریتم کاهش چندجمله‌ای می‌گویند.

معروف‌ترین مسائل NP-hard:

▪ مسئله فروشنده دوره‌گرد

▪ مسئله بزرگ‌ترین خوشه (پیدا کردن بزرگ‌ترین زیر گراف کامل)

کلاس مسائل NP-Complete: به مسائلی گفته می‌شود که هم NP هستند و هم NP-hard. یعنی فضای مشترک دو کلاس NP و NP-hard.

یعنی استفاده هم‌زمان از تأیید زمان چندجمله‌ای جواب مسئله (Verification) و کاهش (Reduction).

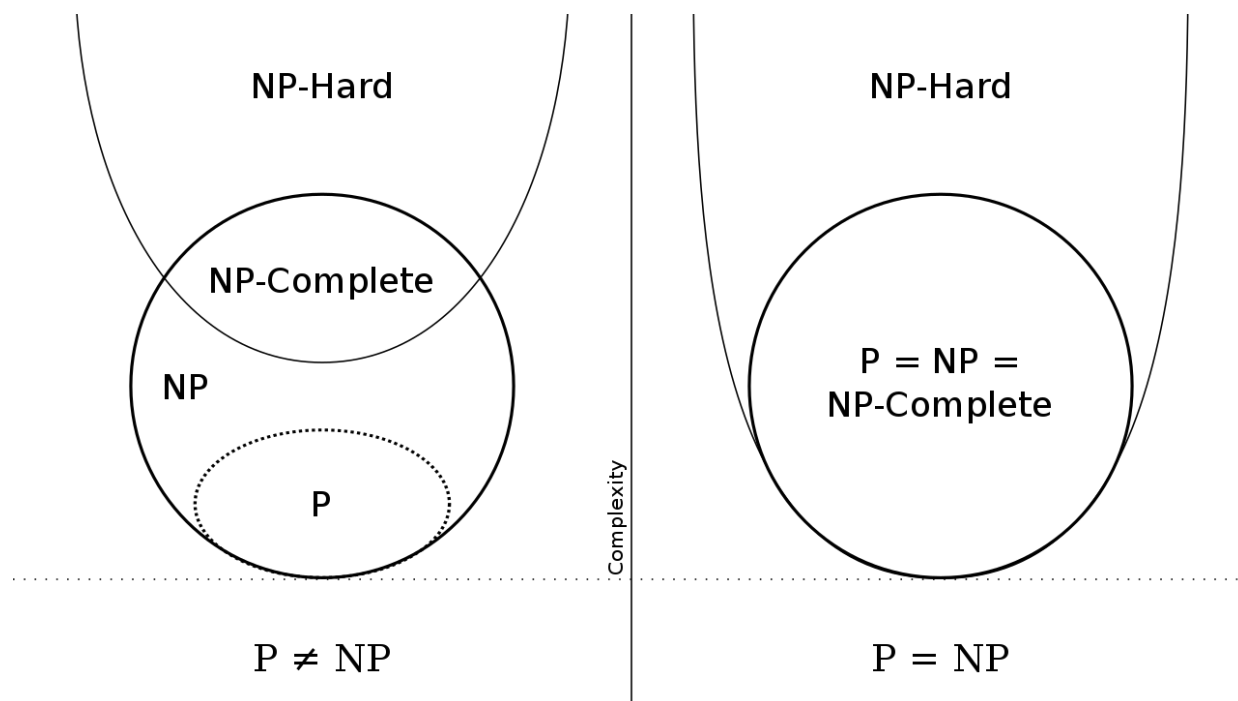
باید توجه داشت که تمام مسائل Np-complete با استفاده از Reduction قابل تبدیل به هم هستند.

از جمله مسائل NP کامل یا NP-Complete می‌توان به مسئله کوله‌پشتی یا پیدا کردن دور همیلتونی در گراف یا فروشنده دوره‌گرد اشاره کرد.

به بیان ساده، NP-Complete ها مسائلی هستند که اثبات شده به سرعت قابل حل نیستند. در تئوری پیچیدگی NP-Complete ها دشوارترین مسائل کلاس NP هستند و جزء مسائلی می‌باشند که احتمال حضورشان در کلاس P خیلی کم است. علت این امر این می‌باشد که اگر یک راه حل پیدا شود که بتواند یک مسئله NP-Complete را حل کند، می‌توان از آن الگوریتم برای حل کردن سریع همه مسائل NP-Complete استفاده کرد. به خاطر این مسئله و نیز به خاطر اینکه تحقیقات زیادی برای پیدا کردن الگوریتم کارآمدی برای حل کردن این گونه مسائل با شکست مواجه شده‌اند، وقتی که مسئله‌ای به عنوان NP-Complete معرفی شد، معمولاً این طور قلمداد می‌شود که این مسئله در زمان – چندجمله‌ای قابل حل شدن نمی‌باشد، یا به بیانی دیگر هیچ الگوریتمی وجود ندارد که این مسئله را در زمان چندجمله‌ای حل نماید. کلاس متشکل از مسائل NP-Complete با نام NP-C نیز خوانده می‌شود.

اگر یک مسئله‌ی بهینه‌سازی ساده باشد مسئله‌ی تصمیم‌گیری مرتبط با آن هم به همان نسبت ساده خواهد بود. همچنین اگر دلایلی داشته باشیم که یک مسئله‌ی تصمیم‌گیری سخت است دلایلی خواهیم داشت که مسئله‌ی بهینه‌سازی مرتبط با آن هم سخت خواهد بود.

مسائل NP در زمان نمایی با توجه به مقدار ورودی اجرا می‌شوند. NP-Complete بدین معنی است که یک مسئله ارزش این را ندارد که سعی شود به صورت بهینه کامل حل شود.



اگر ثابت شود $P=NP$ است، همان‌طور که در شکل سمت راست می‌بینیم پیچیدگی بخش زیادی از مسائل به یک اندازه می‌شود و مسائل NP-hard در بخش بالایی نمودار وجود دارند. ولی تاکنون که $P=NP$ ثابت نشده کلاس‌های پیچیدگی به‌صورت شکل سمت چپ تقسیم‌بندی می‌شود. محور عمودی بیانگر میزان سختی یا پیچیدگی حل مسئله است. در این شکل P زیرمجموعه‌ای از NP هاست. همان‌گونه که مشخص است پیچیدگی و سختی NP ها بیشتر از P است. در بین NP ها بعضی مسائل سخت‌ترین‌اند که در شکل سمت چپ با NP-Complete از آن‌ها یاد شده است. همان‌گونه که در شکل پیداست NP هایی بسیار سخت هم هستند که اشتراک آن‌ها با NP کلاس NP کامل را می‌سازد.