

تاریخ: 1400/03/27

پیوست : ندارد

بسمه تعالی

پاسخنامه آزمون تشریحی الگوریتمهای پیشرفته نیمسال دوم ۱۴۰۰-۱۳۹۹ نام استاد جناب دکتر سید علی رضوی ابراهیمی

نام دانشجو : سید جواد بسیطی - شماره دانشجویی: ۹۹۰۱۸۹۹۸۹

سوال یک میانترم: در معنی و مفهوم علمی الگوریتم بحث نمائید. درس الگوریتم و الگوریتمهای مورد مطالعه برای مهندسين نرم افزار چه کاربرد و فایدهای دارند؟

مفهوم الگوریتم از همان ابتدا که دانشمند بزرگ ایرانی خوارزمی آنرا بنا نهاد، راه حلی سیستماتیک برای حل مسائل است به عبارتی یک توالی صریح، دقیق، بدون ابهام و قابل اجرا به لحاظ مکانیکی از دستورات اولیه است که معمولاً برای انجام کار و هدف خاصی، تعبیه شده‌اند، می‌توان هر مسئله‌ای را با کمک آن به صورت دستورات قابل اجرا بازتعریف کرد. در واقع، یک الگوریتم روال یا فرمولی برای حل یک مسئله بر مبنای انجام یک توالی از فعالیت‌ها است. این نوع فکر پایه و اساس تفهیم و برنامه ریزی ماشین است اصولاً اگر ما نتوانیم بدین شکل سیستماتیک فکر کنیم هرگز قادر به انجام امور مهندسی خود نخواهیم بود. زیرا ترجمه قابل فهم شدن خواسته‌های ما برای ماشین‌ها راه حلی جز سیستماتیک فکر کردن، دارای توالی و دقیق نگاه کردن به مسئله ندارد.

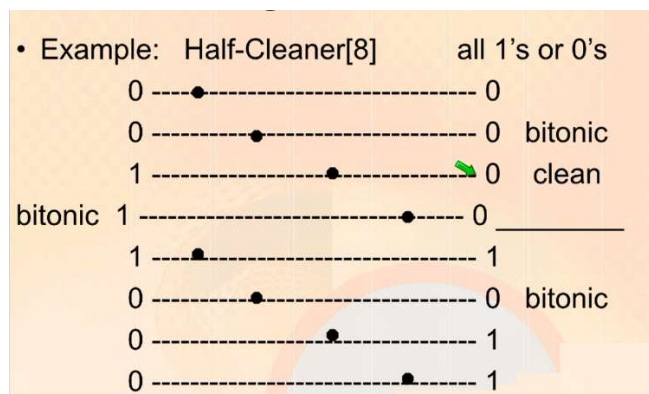
سوال دو میانترم : نحوه عمل نیم پاک کننده در شبکه های مرتب ساز را شرح دهید.

در زمانی که ورودیهای ما بیتونیک (دو آهنگی)، باشند جهت مرتب سازی آنها، در چندین مرحله عمل میشود. که به هریک از این مراحل نیم پاک کننده اطلاق میشود.

نکته مهم اینکه: هر زمان که توالی دو آهنگی، از صفرها و یکها در ورودی نیم پاک کننده اجرا شود، خروجی حاصل شده بدین صورت است که علاوه بر این که هر دو نیمه بیتونیک هستند و حداقل یک نیمه آنها تمیز شده، هر المان در نیمه بالایی به اندازه هر المان در نیمه پایینی کوچک خواهد بود.

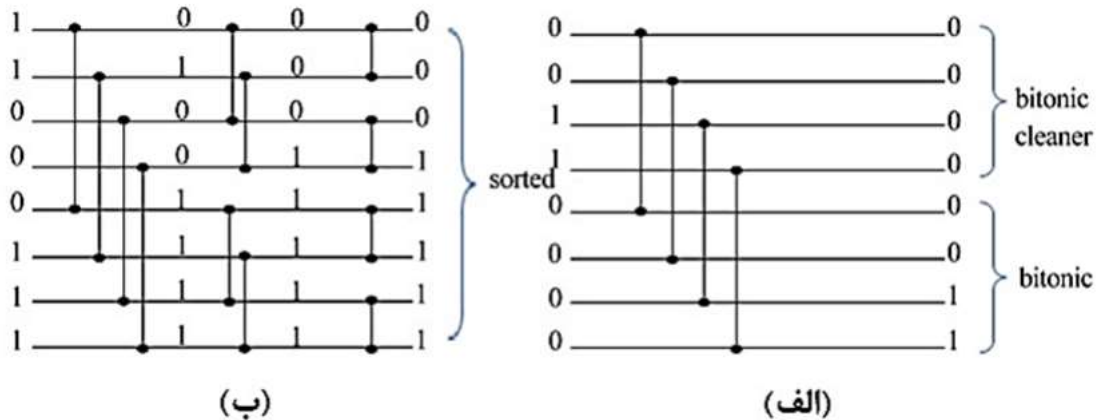
شکل کار بدین صورت است که در شبکه های مقایسه‌ای HALF- CLEANER[n] ورودی برروی خط i با ورودی خط $i+n/2$ مقایسه میشود. فرض بر این است که ورودیها، به صورت بیتونیک یکنواخت افزایشی و یا کاهشی باشند. بعد از پایان مقایسه، حداقل یک نیمه تمیز خواهیم داشت.

مانند شکل زیر که یک توالی دو آهنگی یکنواخت افزایشی را نشان میدهد. همانگونه که مشاهده میشود، بعد از مرحله اول مرتب سازی، در بالا، یک نیم پاک کننده از صفرها و پایین، یک توالی دو آهنگی یکنواخت کاهشی تولید خواهد شد.



نکته: عمق هر نیم پاک کننده، برابر با «یک» بوده و خواهیم داشت: $D(n)=1$

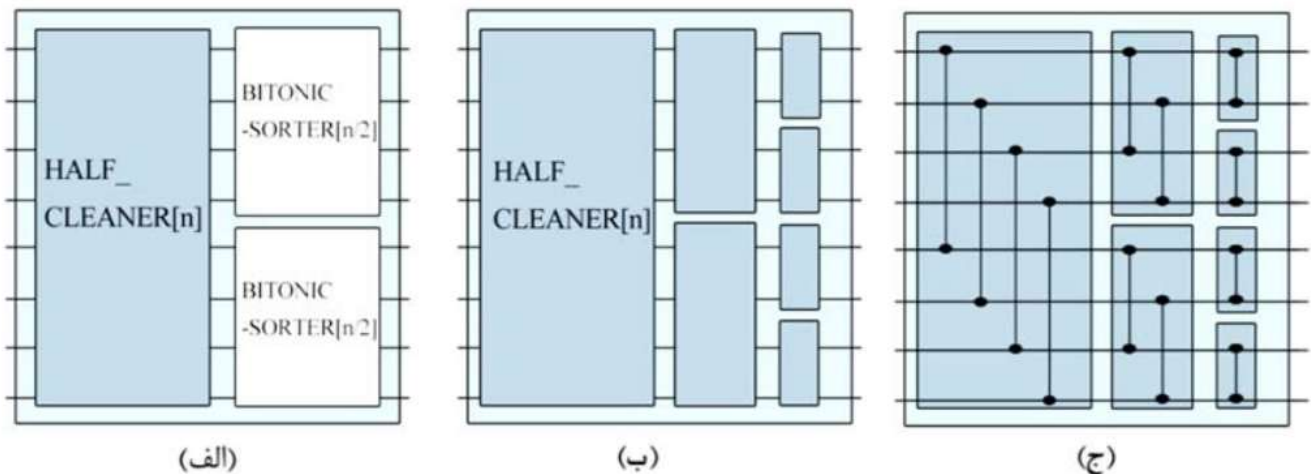
شکل زیر قسمت الف- خروجی توالی شکل گرفته از ورودی صفرها و یک های (00110000) را که به صورت یکنواخت افزایشی است، به روش مرتب سازی دوآهنگی نشان میدهد. همانطور که مشاهده میشود، توالی ورودی، در یک مرحله مرتب شده است. به طوریکه خط i با خط $i+n/2$ مقایسه شده و مقادیر کوچکتر را در نیمه ۲ بالایی، به صورت دوآهنگی پاک قرار میدهد. نیمه پایین نیز دوآهنگی است.



الف) مرتب سازی دوآهنگی برای توالی (00110000)، (ب) شبکه مرتب ساز دوآهنگی با ۸ ورودی، برای توالی (11000111)

* ادامه کار به صورت بازگشتی پیش خواهد رفت تا همه نیمه ها مرتب شده شکل گیرند.

به عنوان مثال در شکل بعدی یک شبکه مقایسه BITONIC- SORTER[n] را وقتی که $n=8$ است، برای توالی (11000111) نشان میدهد. این شبکه به صورت بازگشتی و نیم پاک کننده است که هر مرحله مرتب ساز دوآهنگی، روی نیمی از شبکه است. همانگونه که در ساختار بازگشتی شکل مشاهده میشود، یک توالی در نظر BITONIC- SORTER[n] یک به صورت ابتدا، HALF-CLEANER[n] گرفته شده و پس از انجام مقایسه و جابجایی در هر نیمه - به منظور مرتب سازی -، به HALF-CLEANER[n/2] تغییر میکند. به طوریکه، یکی از نیمه ها پاک است و با قرار گرفتن خروجی دارای مقدار کوچکتر در نیمه بالایی و خروجی با برخی مقدار بزرگتر در نیمه پایینی، نیمه ها مرتب میشود و اگر ورودی نیم پاک کننده، دنباله‌ای دو آهنگی از اعداد دلخواه باشد، آنگاه، هر دو نیمه بالایی و پایینی، دو آهنگی بوده و هر عنصر در نیمه بالایی، حداکثر به کوچکی هر عنصر در نیمه پایینی خواهد بود.



آزمون پایان ترم

سوال یک پایان ترم: نقش الگوریتم برای تأثیرگذاری مهندسی نرم افزار در تحقق شعار سال «تولید، پشتیبانها و مانع - زداییها» چه میتواند باشد؟

باتوجه به تعریف الگوریتم که راه حل سیستماتیک مسئله است ابتدا دو باره آنرا مرور کنیم و به عبارتی یک توالی صریح، دقیق، بدون ابهام و قابل اجرا به لحاظ مکانیکی از دستورات اولیه است لذا این مسئله را نیز قطعاً صرفاً با شعار و حرف نمی توان حل نمود و باید به راه حلی سیستماتیک پایبند بود. طبق تجربه ای که اینجانب چندین سال با آن سرو کار داشته ام به این نتیجه می رسم که باید به صورت سیستماتیک توالی را در پیش گرفت که ما را به سمت هدف رهنمون کند و صرفاً جسته و گریخته انجام اقداماتی پراکنده که آنهم فقط مجری آن نمایشی برگزار کی کند هدفی را تأمین نمی کند شعار هر سال نیز نقشه‌ی راهی ست برای حل مشکلات موجود و رسیدن به برنامه‌ی کلان کشور. امروزه تولید نرم افزار و خدمات دیجیتال بخش قابل توجه ای از تولید هر کشور را به خود اختصاص می دهد. برهیچ کس برنامه نویسی پوشیده نیست که برنامه نویسی و تولید نرم افزار یعنی برطرف کردن نیازی با روندی آزموده شده، که همان الگوریتم است. پس برای تحقق این شعار باید الگوریتمی عمل کرد.

به عنوان مثال به صورت الگوریتمیک روال زیر برای تحقق این شعار اتود می گردد که البته در جاهایی پیش نیاز و هم پوشانی هایی برای جواب سوال دو نیز دارد:

- مرحله اول: بررسی همه جانبه شعار سال و تنظیم هدف در هر ارگانی و یا گروههای صنعتی و بازاری
تهیه نقشه راه: برای آن نقشه همه در بدنه نظام و جامعه باید کمک کنند و مواردی که لازم می بینند را یک به یک برای کمیته تهیه کننده نقشه راه تفهیم کنند (مثلاً واكشی قوانین غلط و نادقیق و غیرو) که البته در سالهای دور تا کنون نقشه راه های فراوانی ترسیم شده ولی در اجرا پیاده سازی نشده به همین منظور یک راست به بخش اجرا می روم
- مرحله دو: در اجرا همه یک به یک مراحل الگوریتم باید اجرا شود و در عمل حذف و یا بیرمق و کمرنگ کردن بعضی و اجرای بعضی دیگر با هر توجیهی کل الگوریتم را مختل می کند. مثلاً به یاد دارم سالها پیش "قانون بهبود مستمر محیط کسب و کار" بعد از سالها تلاش اساتید ما در جاهایی مانند سازمان مدیریت صنعتی و غیره تصویب شد ولی هیچ گاه حتی جلسات مستمر آن با استادان و غیره اجرایی نشد و اهمیت داده نشد انگار که هر عضوی از این ماشین مملکت خود می تواند تصمیم بگیرد که چه چیزی قابل اهمیت دادن هست و یا چه چیزی نیست! خوب تا زمانی که ما با این بدنه خود رای سرو کار داریم کاربرد الگوریتم به جایی نمی رسد، راه حل آن:

پس داشتن یک مرحله صفرم: حذف مدیران خود رای و آنها که ذهن خود و خط فکری خود را در راستای قوانین خواسته شده قرار نمی دهند. به عبارتی اولین مرحله: **ایجاد فرهنگ حاکمیت قانون** تا اول قانون را که اصل الگوریتم اجرایی ماست حاکم نماییم و افراد پناهی غیر قانون را در جایگاههای خود بدانند الگوریتم ما پیش نخواهد رفت. (همین یک نکته که مثلاً حتی رهبر کل جامعه را بالاتر از قانون بدانیم خود این موضوع به این معنی است که روال سیستماتیک که قانون شکل داده را در جایی می توان با تأثیر گذاری ذهنی بر روی گروهی نزدیک به وی و بعد انتقال آن به سیستماتیک اجرایی به بدنه ترسیمی قانون و الگوریتم در جایی می توان ترمز وارد کرد .)

به هر حال متأسفانه در این مسیر ما با ماشین صرفاً سرو کار نداریم طرف ما یکسری آدمها هستند که باید موارد را اجرا کنند و نقطه تمایز شون با ماشینها اینکه امکان تظاهر به اجرا دارند در حالیکه در پشت و خفا اصلاً معتقد به آن نیستند.
جملاتی را در این ۲۵ سال کار در صنعت به یاد دارم:

- ۱- جمله اول را رده بالاترین مدیر کاداستر کل کشور در سال ۸۹ به من زد، ما بر روی دریای نفت خوابیده ایم و نیاز به اشتغالزایی داریم نه بهره وری سیستم نرم افزاری شما در راستای بهره وری حرکت می کند و طبعاً منجر به کاهش نیرو است پس نیازی به این چنین ابزارهایی نیست. (نفرتین نفت)

۲- عذرخواهی کنم از جنابعالی که شاید پاسخ این دو سوال را شاید به شکلی که مد نظر جنابعالی بود نیاوردم زیرا سالهاست که بیاد دارم روشها برنامه ها و الگوریتمهای پیشنهادی و نقشه راه ها و غیره را دیده ام ولی به یاد دارم این شعر را که وقتی "خانه در جای دیگری ویرانست، خواجه در فکر نقش ایوان است" (کلمه "پایبست" به عمد در شعر تغییر داده شد، چون نخواستیم پایه ای چیزی را زیر سوال ببریم) در دوران تحصیل MBA جمله ای از انیشتن بسیار مرور میشد به نقل مضمون که: ما در امور خود گرفتار مشکلات و مسائلی می شویم که تفکرات و خط فکری خود ما ایجاد کرده است. این را در برنامه نویسی بسیار با آن برخورد کردم وقتی ساعتها گرفتار باگی در برنامه می شدم باید به عقب بر می گشتم و با نگاه کلی تر به موضوع می دیدم باگ اصلا در آنجا که ما فکر می کنیم یا خط فکری ما به سمت آن رفته نیست و گرنه رفع آن زمانبر نمی شد اینکه ما گرفتار خط فکری غلط خود شده بودیم و تمرکز را در جایی گذاشته بودیم که اصلا این باگ در تفکرات دیگر ما که فکر می کنیم درست است روی داده و ما از آن خطوط فکری دور هستیم دقیقا موضوعات مورد بحث جنابعالی در دو سوال اخیر نیز فکر می کنم در این معقوله است اینکه من برای افزایش تولید و یا ایجاد گرایش دانشجویان به سمت برنامه نویسی بخواهم برنامه یا الگوریتم بدهم شاید فرو رفتن در جزئیات باشد، زنجیره تولید (CKD) و الگوریتمهای فراوان خود را دارد ما باید الگوریتمی برای اصلاح کلان تر امور پیشنهاد کنیم این موارد خود به خود توسط فعالان حوزه ها حل خواهد شد. چون خود می دانند اگر حل نکنند بازار کار آنها را بد جور تنبیه خواهد کرد و حتی در مورد اخیر نیز دانشجویان خود متوجه خواهند شد بازار کار به دنبال چی از آنان است. ولی در حال حاضر که منطق بازار حاکم نیست و مثلا با دولتی بودن اقتصاد بودجه ریزی به شکلی است خارج از منطق بازار و اصولا هر منطق دیگری به جز رابطه بازی، مثلا من به جای اینکه دنبال تولید یک محصول نرم افزاری بهره ور باشم باید دنبال خط بودجه سازمانی باشم که آشنایانی نیز جهت برنده شدن مناقصات در آن دارم و با ایجاد صمیمیتی قوی به همراه سود دوطرفه بلکه سهم بیشتر برای او کسب درآمد کنم، (وقتی قصد کردم با این روش مبارزه کنم و در مسیر آن شنا نکنم فقط طی دوسال کاملا از بازار حذف شدیم).

خوب پس شاید به نظر برسد مبارزه با فساد اولین قدم در الگوریتم ماست ولی در این بخش نیز ما مراحل داریم و آن اولین قدم مبارزه با فساد از درون خود مدیر بالادست مرتبه یک باید شروع شود آیا مدیر بالادست حاضر است به جای کار کردن با اقارب و هم حزبیهای خود از همان معاونین درجه اول، بر اساس تخصص و تجربه و سواد انتخاب معاون و افراد مجموعه کند نه بر اساس چیز دیگری این اولین مرحله حل الگوریتمیک ماست و گرنه مدل انتخاب غلط به طور بازگشتی در کل سیستم جاری می شود. تا کوچکترین جزء سیستم به یاد دارم وقتی با کوچکترین سرپرست جزء سیستم نیز کار می نمودیم و او را نیز فارغ از دغدغه پیشبرد امور می دیدیم به شکلی که بودن خود را در این پست نه بر اساس امور پیشبرده بلکه موارد دیگری می دانست.

خوب چگونه دغدغه پیشبرد امر را در وی تزریق می نمودیم؟ چون در جایگاه آن نبودیم و توجیه مقام بالاسری نیز بد گویی ما تعبیر می شد. لذا این الگوریتم پیشنهادی از درون مدیر اول باید آغاز گردد:

- ۱- تهیه مستندات واقعی در همه رشته ها که ابتدا تعریفاتی از تخصص و تجربه و سواد ارائه دهد.
- ۲- اجرای دقیق قانون و توجه به روح قانون: صرفا پناه قراردادن قانون برای همه آحاد جامعه حتی بالاترین مدیر رده یک به جز قانون پناه دیگری برای حفظ خود احساس نکند. (به طور همزمان تکمیل ابهام زدایی و حذف موارد دست و پاگیر و کلا تصحیح قانون کنونی پیش خواهد رفت ولی در شروع از همین قوانین موجود خواه غلط باید مدد جست).
- ۳- فعال سازی گروههای واقعی صنفی که از متخصصان را از درون خود بالا بیاورند. و های لایت کنند. (در همه گروهها از گروههای تولیدی و صنعتی گرفته، بازاریان و تا و یا حتی زنان خانه دار) این موارد را بعدا مدیر بالادستی برای انتصابات خود دستچین می نماید.

۴- چون قانون پناه فرد شماره یک است پس نگران توطئه همکاران نیست لذا معاونین و همکاران خود را از موارد ردیف سه دستچین می نماید نه از اقارب خود، بدین شکل او قادر خواهد بود نهایت مدیریت خود را بر فرد اعمال کند ، در حالیکه بر اقارب خود هرگز چنین توانی ندارد.

۵- موارد سه و چهار در کل هرم سازمانی به صورت بازگشتی (ریکرسیو) امتداد می یابد .

۶- به همین ترتیب راه اندازی دستگاه نظارتی بسیار قوی آنهم صرفا بر مبنای قانون نه روابط و یا مصالح اندیشی در قوه دیگر متاسفانه مصلحت اندیشی و پیروسی است که به جان اجرای قانون در مملکت افتاده و همه جا به جای اجرای مو به مو قانون و مجازات متخلفین آن نگاه به سوابق کاملا نامرتبب فرد شده (مثلا حضور در جبهه، اخلاص و تعهد نمایان نموده و ...) و لاجرم توان مجازات را از دستگاههای نظارتی گرفته است و در صورت عدم شفافیت قانون و یا دست و پا گیری و یا ابهام داشتن آن باید پروسه جدا گانه و موازی برای اصلاح آن آغاز کرد.

انتخاب ناظرین مهمترین عامل است که به فرض اگر در فلان نهاد مجری بخش صنعتی ، از فعالین صنعت انتخاب شد از صنف مصرف کنندگان آن در کنار آن یا در نهاد نظارتی آن قرار گیرد.

۷- تشکیل تیم پایش بقیه ایرادات فرعی موجود در موانع تولید با امداد از همان گروههای صنفی و جمع آوری موانع تولید و بررسی متمرکز آنها

۸- ارسال لوایح تصحیح قانونی در مواردی که احتیاج به تصحیح قانون دارد

۹- ادامه از مرحله ۲

سوال دو پایانترم: با ذکر مثال و ارائه یک الگوریتم اولیه) لزومی به اثبات درستی الگوریتم ارائه شده وجود ندارد(تالش نمائید قدمی برای حل مشکل کارآمدی دانشجویان مهندس برای مشارکت در تولید نرم افزار در دوران دانشجویی و یا پس از آن ارائه نمائید؟ (استفاده از الگوریتمهای مطالعه شده در درس مورد استقبال خواهد بود)

مرحله اول آنالیز مسئله: در زمان تحقیق راجع به مشکل برخورد می کنیم با موارد متعددی که مانع الفت دانشجویان با تولید شده ، اولین مشکل به نظر من وجود برخی اساتیدی است که صرفا به تئوری موضوعها عادت نموده و علاقه وافر صرفا به جنبه تئوری ها دارند به عبارتی خود در بازار کار وجود نداشته و انس و الفت و همگرایی با آن نداشته اند پس مرحله اول اجرا پاکسازی و یا تغییر نگرش برخی اساتید که متاسفانه بر خلاف جنابعالی یا خود اصلا در زمینه تولید وارد نشده اند و یا علاقه و الفتی با آن نداشته و طبعاً آنرا در شیوه ها و تدریس خود خواه نا خواه موثر می کنند.

در مرحله بعدی نگاه سود و زیان و درآمد زایی و ارتباط مستقیم دادن آن با درآمد زایی دانشجویان عموماً برای هزینه تحصیل و زندگی خود مشکل دارند و همچنین دانشگاهها که برای تامین بودجه خود اگر وابسته به صنعت باشند یک طرف قضیه و در طرف دیگر صنعت نیز از یک طرف تشنه انجام کار درست و ارزان است اگر تا کنون این دو بهم پیوند نخورده اند یکی همان همان نفرت باشد که سالهاست گریبان ما را گرفته است (ادامه ارجاع به پاسخ سوال یک که سعی شده سیستماتیک امور را مرحله بندی کند).

مرحله سه : با تحقق موارد پاسخ در سوال یک و بهادار شدن صنعت و پویا شدن آن و لاجرم شدن تامین بودجه دانشگاهی از تولید و صنعت الگوریتم ما خود به خود از خود نفرت جدا شده و ادامه می یابد.

سوال سه پایان ترم: دسته های سختی مسایل و تفاوت آنها را تشریح نمایید.

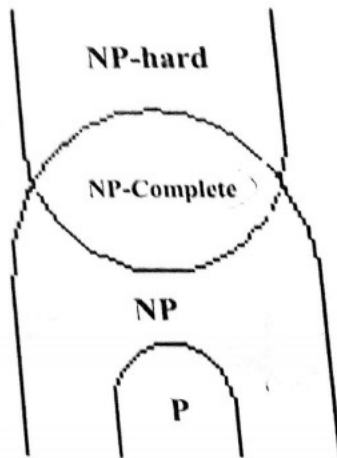
سختی در ذات مسئله نهفته است به طور کلی مسائلی که با الگوریتم های زمان چند جمله ای قابل حل هستند، حل پذیر یا آسان هستند و مسائلی که به زمان فرا چند جمله ای نیاز دارند حل ناپذیر یا سخت نامیده می شوند. نظریه هایی وجود دارد که مسائل را بر اساس دشواری آنها دسته بندی می کند این دسته بندی معروفترین شان به نام نظریه پیچیدگی محاسباتی شامل کلاسهای پیچیدگی زمانی زیر است:

۱- P

۲- NP

۳- NP-Complete

۴- NP-Hard



رابطه بین این کلاسها را تا حدودی به شماتیک روبرو می توان ترسیم کرد:

کلاس P: هر مسئله ای که بتوان در زمان چندجمله ای حل کرد، به کلاس P تعلق دارد. برای

این مسائل یک الگوریتم قطعی وجود دارد. و در بدترین حالت به ازای k ثابت در زمان $O(n^k)$ حل می گردد ولی در کلاسهای NP مسایلی که خودش شاید در زمان چندجمله ای حل نشود، ولی اگر یک راه حلش را داشته باشیم، می توانیم درستی آن را در زمان چندجمله ای واریسی کنیم. یک مسئله در صورتی در کلاس NP-Complete قرار دارد که در NP نیز باشد. و بطوریکه، هیچ الگوریتم شناخته شده و قابل اجرایی، با زمان چندجمله ای برای آنها وجود نداشته باشد. به عبارت دیگر، پیچیده ترین مسائل کلاس NP در کلاس NPC وجود دارد. به عبارتی مسائل NPC جزء سخت ترین مسئله ها در کلاس NP هستند. ویژگی این مسائل، استفاده از مسئله های دیگر کلاس NP، برای حل آنها با روش کاهش است. همچنین عدم وجود یک الگوریتم چند جمله ای شناخته شده ای برای حل آنها جزء ابهامات فعلی علم است.

کلاس NP-Hard: اینها مسائلی هستند که نه تنها خودشان در چند جمله ای حل نمی شوند، بلکه راه حلشان هم شاید در چند جمله ای قابل واریسی نباشد. مسایل NP-Complete، این پی سخت هم هستند ولی مسایلی در NP-Hard هست که سخت تر از بقیه است و NP-Complete نیست. اما بخاطر سختی حل آنها از روش های دیگری استفاده می شود. مثلاً یکی از این روش ها الگوریتم های تقریبی هستند. ویا الگوریتم های ابتکاری و تصادفی. همه این روش ها یک مسئله NP-Hard را دقیق حل نمی کنند، ولی مزیتشان اینست که جواب تقریبی نزدیک به دقیق را خیلی زود و تند برای ما تولید می کنند.

همچنین کلاس CO-NP: مسائلی هستند که متمم آنها متعلق به NP است اما خودشان را نمی دانیم متعلق به NP هستند یا خیر.

با تشکر از حوصله جنابعالی - سید جواد بسیطی