**میانترم**

1-هر علمی یک هدفی دارد، مهم این است که بفهمیم الگوریتم به چه معنی است؟

الگوریتم یکی از راه‌های حل مسئله با روش‌های ریاضی و منطقی است، این روش برای نخستین بار توسط خوارزمی (دانشمند ایرانی) مورد توجه قرار گرفت. به بیان صحیح تر روش سیستماتیک حل مسئله. سیستماتیک به این معنا که نقطه شروع و پایان داشته باشد، شفاف باشد، تکرارپذیر باشد و علاوه بر تکرارپذیربودن باید به صورت عملی هم امکان‌پذیر باشد و هم بهره ور باشد، یعنی باید بتواند در زمان معین و با هزینه قابل قبول مسئله را حل کند. روش ومتد با الگوریتم دارای دو معنی متفاوت هستند. متد یا شیوه، روش نگاه به مسئله است یعنی سمت و سو؛ الگوریتم گامهای حل مسئله است یعنی دقیق‌تر و عملیاتی‌تر و به واقعیت نزدیکتر است. متد کلی‌تر و الگوریتم دقیق‌تر و ریزبین تر است. حال این سؤال پیش می‌آید که آیا برای هر مسئله‌ای الگوریتم خاص خودش وجود دارد؟ مثلاً اگر در دنیای واقعی 1000 تا مسئله داریم آیا برای حل آن 1000 تا الگوریتم وجود دارد؟

برای بعضی از مسائل چند تا راه‌حل (راه‌حل باید عملیاتی باشد) وجود دارد، بعضی از مسائل ممکن است راه‌حل داشته باشند ولی ما پیدا نکرده‌ باشیم و بعضی از مسائل ممکن است راه‌حل نداشته باشند در واقع راه‌حل عملیاتی نداشته باشند. خیلی از مسائل به هم شبیه هستند ممکن است ظاهراً شبیه نباشند ولی باطن آنها شبیه است و اگر ما یکی از آنها را، حل کنیم باقی مسائل را می توان حل کرد. ما مسائل را دسته‌بندی می‌کنیم، آنها را انتزاع (جزء) می‌کنیم و سپس اجزاء مشابه را پیدا می‌کنیم، به مدل ریاضی تبدیل می‌کنیم و مدل ریاضی را حل می‌کنیم. ما باید یاد بگیریم که الگوریتم، ساختمان داده، گرافها انتزاعی است از مسائل مختلف که در جاهای مختلف مطرح می‌شود. هرجا که مسئله هست باید الگوریتم پیدا کنیم، راه‌های ابتدایی وجود دارد ولی تکرارپذیر و عملیاتی نیست. باید به مسائل دید سیستمی پیدا کرد. اگر الگوریتم بدانیم راه‌حل‌های بهتری می‌دهیم اگر یک مسئله‌ای در دنیا حل نشده شاید حل نشدنی باشد ولی باید روی مسائل کار شود. به بیانی دیگر، می‌توان گفت که الگوریتم یک مجموعه از عملیات مرتب و محدود است که باید به منظور حل مسئله خاصی دنبال ‌شوند. این یعنی یک زنجیره از دستورالعمل‌های دقیق وجود دارند که باید به ترتیب خاصی دنبال شوند. هدف الگوریتم حل یک مسئله است و این یعنی الگوریتم‌ها دارای هدف از پیش تعریف شده هستند. در واقع، هدف از نوشتن یک الگوریتم، تولید یک خروجی است.

اگر الگوریتم‌ها برای مسائل جهان واقعی نوشته شوند، باید ضمن نوشتن آن‌ها، کلیه حالاتی که ممکن است هنگام اجرای الگوریتم با آن مواجه شد، در نظر گرفته شوند. به همین دلیل است که الگوریتم‌هایی که برای مسائل دنیای واقعی نوشته می‌شوند، پیچیده، بزرگ و همچون درختی از دستورالعمل‌ها هستند. اما دنیای تکنولوژی انبوهی از الگوریتم‌ها را در اطراف ما قرار داده که معمولاً بدون این‌که به آن‌ها توجه کنیم، از نتایج آنها استفاده می‌کنیم. در اینجا از چند نمونه نام می‌بریم:

* الگوریتم‌های مورد استفاده در [موتورهای جستجو](https://motamem.org/%D9%85%D9%88%D8%AA%D9%88%D8%B1%D9%87%D8%A7%DB%8C-%D8%AC%D8%B3%D8%AA%D8%AC%D9%88/) برای پیدا کردن بهترین پاسخ‌ها برای پرسش کاربران
* در شکل کلی‌تر، [الگوریتم‌های پیشنهاددهنده](https://motamem.org/%D9%85%D9%88%D8%AA%D9%88%D8%B1%D9%87%D8%A7%DB%8C-%D9%BE%DB%8C%D8%B4%D9%86%D9%87%D8%A7%D8%AF%DA%A9%D9%86%D9%86%D8%AF%D9%87/) که در بسیاری از نرم‌افزارها از جمله [شبکه های اجتماعی](https://motamem.org/%D8%B4%D8%A8%DA%A9%D9%87-%D9%87%D8%A7%DB%8C-%D8%A7%D8%AC%D8%AA%D9%85%D8%A7%D8%B9%DB%8C-%DA%86%D9%87-%D9%87%D8%B3%D8%AA%D9%86%D8%AF-%D9%88-%DA%86%DA%AF%D9%88%D9%86%D9%87-%D8%AA%D8%B9%D8%B1%DB%8C%D9%81/) و [خرده فروشی](https://motamem.org/%D8%AE%D8%B1%D8%AF%D9%87-%D9%81%D8%B1%D9%88%D8%B4%DB%8C-%DA%86%DB%8C%D8%B3%D8%AA-%D9%88-%D8%A2%DB%8C%D8%A7-%D8%AA%D8%B9%D8%B1%DB%8C%D9%81-%D8%AE%D8%B1%D8%AF%D9%87-%D9%81%D8%B1%D9%88%D8%B4%DB%8C-%D8%AA/) های دیجیتال (مثل[آمازون](https://motamem.org/%D8%B4%D8%B1%DA%A9%D8%AA-%D8%A2%D9%85%D8%A7%D8%B2%D9%88%D9%86/) و دیجی کالا)‌ دائماً از آنها استفاده می‌کنیم.
* [الگوریتم‌های رمزنگاری](https://motamem.org/%D8%B1%D9%85%D8%B2%D9%86%DA%AF%D8%A7%D8%B1%DB%8C-%DA%86%DB%8C%D8%B3%D8%AA-%D8%A7%D9%84%DA%AF%D9%88%D8%B1%DB%8C%D8%AA%D9%85-%D9%86%D8%A7%D9%85%D8%AA%D9%82%D8%A7%D8%B1%D9%86/) که می‌کوشند امنیت تبادل اطلاعات در شبکه‌های ارتباطی را افزایش دهند.
* الگوریتم‌های فشرده سازی اطلاعات که حجم داده‌ها را تا حد امکان کاهش می‌دهند.

امروزه بیشترین کاربرد الگوریتم در برنامه نویسی کامپیوتر است. در دنیای امروز مسائل فراوانی وجود دارد که می‌توان توسط [الگوریتم‌ها](https://fa.wikipedia.org/wiki/%D8%A7%D9%84%DA%AF%D9%88%D8%B1%DB%8C%D8%AA%D9%85) راه حل‌های مناسب و بهینه‌ای برای آن‌ها ارائه نمود. استفاده از الگوریتم‌ها در این مسائل باعث صرفه جویی در وقت و هزینه شده و راهکارهای نوینی را پیش رو قرار می‌دهد.الگوریتم به مهندس نرم‌افزار برای رسیدن به محصولی با کیفیت بالا در تولید نرم افزارکمک می کند. راه حل مسائل پیچیده و بزرگی که بر سر راه یک مهندس نرم‌افزار پیش می‌آیند، تسلط بر الگوریتم‌هاست؛ زیرا با نوشتن یا تصور یک الگوریتم منطقی می‌توان بر تمام مسائل غلبه کرد. استفاده از الگوریتم‌ها در کاهش هزینه‌های مالی و زمانی یک پروژه نیزاهمیت زیادی دارد. الگوریتم‌ها دارای نقش مهمی در [برنامه‌نویسی](https://7learn.com/tutorials/programming) و حل مسئله هستند و از لحاظ کارایی و با توجه به نوع مسئله انواع مختلفی دارند. از جمله الگوریتم‌های پرکاربرد که حتی در زندگی واقعی ما استفاده می‌شود الگوریتم‌های مرتب‌سازی می‌باشند.

2-در این سوال ابتدا به توضیح شبکه مرتب‌سازی Bitonic می‌پردازیم.

Bitonic یکی‌ از الگوریتم‌های مرتب‌سازی برای ماشین‌های موازی است که به عنوان یکی از بهترین الگوریتم‌های مرتب‌سازی از نظر پیچیدگی و هزینه شناخته می‌شود. یک سری مرتب شده، یا سری صعودی است یا سری نزولی اما سری بایتونیک ترکیبی از دو زیر سری صعودی و نزولی است. اولین قدم در ساخت یک شبکه مرتب ساز کارآمد ، ساخت یک شبکه مقایسه است که بتواند هر دنباله بایتونیکی را مرتب کند: توالی ای که به صورت یکنواخت افزایش می یابد و سپس به صورت یکنواخت کاهش می یابد ، یا اینکه یکنواخت کاهش می یابد و سپس یکنواخت افزایش می یابد. در مرتب سازی Bitonic ما از دو مرحله استفاده می‌کنیم، مرحله اول تبدیل اعداد نامرتب به یک توالی Bitonic و سپس مرتب‌کردن توالی Bitonic به اعداد مرتب شده. بنابراین طبق این گفته همیشه اعداد ما باید توانی از 2 باشند.

* در ابتدا به این می‌پردازیم که چطور رشتة بایتونیک بسازیم؟

این عمل به صورت چرخشی انجام می‌شود. مثلاً اگر یک رشته 8 تایی داریم آن‌را به دو رشته 4 تایی تقسیم می‌کنیم. یعنی الان دوتا رشته 4تایی داریم که جداگانه برروی آن کار می‌کنیم. رشته 4تایی در حقیقت دوتا 2تایی است. زمانی این 4تایی بایتونیک می‌شود که دوتای اول را نزولی و دوتای دوم را صعودی سورت کنیم. دراین صورت یک 4تایی بایتونیک داریم. برای 4تایی دوم هم همین کار را تکرار می‌کنیم.

* حال باید ببینیم که چطور رشته بایتونیک را سورت کنیم؟

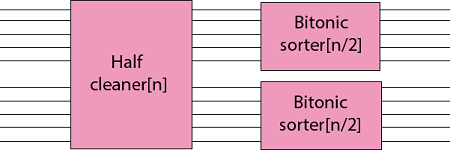
ما یک رشته بایتونیک 4تایی ساختیم(یعنی نصفش نزولی و نصفش صعودی بود). نصف چهار می‌شود دو.

ما آیتم اول را با آیتمی که دوتا بعد از خودش هست مقایسه می‌کنیم. یعنی آیتم یک را با سه و آیتم دو را با چهار. آیتمها را به صورت نزولی مرتب می‌کنیم. 4تای دوم هم با انجام همین مراحل سورت می‌کنیم منتها به صورت صعودی. این توالی یک رشته بایتونیک 8تایی می‌شود. حال طبق این الگوریتم این رشته بایتونیک 8تایی را هم سورت می‌کنیم.

به عنوان مثال ، توالی های 1 ، 4 ، 6 ، 8 ، 3 ، 2 و 9 ، 8 ، 3 ، 2 ، 4 ، 6 هر دو بایتونیک هستند.

Half-Cleaner (نیم‌پاک کننده):

یک مرتب‌کننده بایتونیک از چندین مرحله تشکیل شده است که به هر مرحله نیم‌پاک‌کننده گفته می‌شود. هر نیم‌پاک‌کننده یک شبکه مقایسه‌ای از عمق 1 است که در آن ورودی خط i با خط i + n / 2 برای i=1,2,…,n/2 مقایسه می‌شود. (ما فرض می کنیم که n زوج است).



هنگامی که یک توالی بایتونیک از 0 و 1 به عنوان ورودی به یک نیم‌پاک‌کننده وارد می‌شود ، نیم‌پاک کننده یک توالی خروجی تولید می‌کند که در آن مقادیر کوچکتر در نیمه بالا هستند و مقادیر بزرگتر درنیمه پایین و هر دو نیمه بایتونیک هستند و حداقل یکی از نیمه‌ها تمیز است.

در شکل زیر دو نمونه متفاوت صفر - یک ورودی و مقادیر خروجی نشان‌داده شده‌است. فرض می‌شود که ورودی بایتونیک باشد. یک نیم‌پاک‌کننده تضمین می‌کند که هر عنصر خروجی نیمه بالایی حداقل به اندازه هر عنصر خروجی نیمه پایینی کوچک است. علاوه‌براین، هر دو نیمه بایتونیک هستند و حداقل یک نیمه تمیز است.

شکل زیریک نیم‌پاک‌کننده با 8 ورودی و 8 خروجی را نشان می دهد.

Diagram

Description automatically generated

بنابراین اگر ورودی نیم‌پاک‌کننده یک توالی بایتونیک از صفر و یک باشد، آنگاه خروجی دارای خواص زیر است:

1-هر دو نیمه بالایی و پایینی بایتونیک هستند

2-هر عنصر در نیمه بالا حداقل به اندازه هر عنصر نیمه‌پایینی کوچک است و حداقل یک نیمه تمیز است.

**پایان‌ترم**

1-ما به عنوان یک مهندس نرم‌افزار باید یک سری راهکارها و راه‌حل‌های سیستماتیک ارائه دهیم تا بتوانیم کمکی به رونق تولید و رفع موانع تولید کرده باشیم در ابتدا به صورت کلی مشکلات موجود رابررسی نموده و راهکارهای کلی ارائه می‌دهیم و سپس این راهکارها را در یک محیط کوچک پیاده‌سازی می‌کنیم. حال مسائل و مشکلاتی که در دنیای واقعی بر سرراه تولید داریم به طور خلاصه بیان می‌کنیم. پشتیبانی و مانع زدایی از تولید، اولویت اصلی برای رشد و توسعه اقتصادی کشور است،  دراین میان نقش و اثربخشی فعالیت مدیران اجرایی برای رفع موانع موجود و هدایتگری بخش تولید در راستای تحقق این شعار محوری بسیار مهم و قابل توجه است و البته کار و تلاش شبانه روزی بر مبنای تعهد، تخصص، صداقت و حمایت و دلسوزی منطقی لازمه دستیبابی به  این مهم برای رونق تولید و توسعه اقتصادی کشور و در نهایت بهبود اوضاع معیشتی و رفاه بیشتر مردم است. تحقق این مهم علاوه بر حمایت مالی و قضایی از فعالیت بخش خصوصی و تولید کنندگان به حمایت های علمی و تخصصی نیز نیاز دارد تا با برگزاری دوره های آموزشی و علمی هدفمند، زمینه تحقق شعار سال برای رونق و جهش تولید بیش از پیش فراهم شود. استفاده از توان علمی و ظرفیت مراکز دانشگاهی، پژوهشی و فنی و حرفه ای به عنوان حامی بخش تولید و تولیدکنندگان نیز به عنوان یک ضرورت باید در دستور کار مدیران اجرایی کشور قرار گیرد. رفع مشکلات مالی و نقدینگی واحدهای تولیدی از دیگر اقدامات موثر برای بهبود فعالیت های اقتصادی در راستای تحقق این شعار می‌باشد و همچنین بانک ها و موسسات مالی باید با تسهیل فرایندها و رفع قوانین دست و پاگیر، به صورت جدی و عملی وارد میدان حمایت از تولیدکنندگان شده و با تامین منابع و پرداخت تسهیلات موردنیاز، بستر مناسب برای شکوفایی فعالیت واحدهای تولیدی، اقتصادی و صنعتی کشور را فراهم کنند. همچنین حمایت دستگاه قضایی از تولیدکنندگان در راستای رفع موانع تولید امری مهم است. مشکلات بروکراسی اداری یکی از مشکلات واحدهای تولیدی بوده و هزینه های تولید آنها را نیز افزایش می دهد. این درحالی است که برای واردات به کشور تسهیلات بسیار بیشتر است و توزیع کنندگان ترجیح می دهند از کالاهای خارجی استفاده می کنند بنابراین اگر می خواهیم جهش تولید داشته باشیم باید این نوع مسائل برای تولید کنندگان حل شود. توسعه اقتصادی مستلزم حمایت از سرمایه‌گذاران و فعالان عرصه کار و تولید است. مثلا در کشاورزی **"سرمايه و دانش"** دو عامل اساسي براي تحقق جهش توليد**در كشاورزي مي باشد كه متأسفانه در سال هاي گذشته  جريان ورود هر دو عامل به محيط كشاورزي همواره ضعيف و كند بوده است؛ به طوري كه علي رغم وجود سرمايه هاي فراوان مردمي و نقدينگي در كشور، شاهد ورود حداقلي از سرمايه هاي مردمي به فعاليت هاي كشاورزي در مقايسه با ساير فعاليت هاي اقتصادي مي باشيم. بررسي‌ها نشان می‌دهد بخش عمده اي از اين عدم رغبت و جذابيت براي ورود سرمايه هاي مردمي مربوط به سختي‌ها و پيچيدگي‌هاي اخذ**مجوز كسب و كارهاي كشاورزي مي باشد كه در اين خصوص **آسان سازي صدور مجوزها مي تواند منجر به تقويت جريان ورود سرمايه مردم به محيط كشاورزي گردد.** شناسايي موانع پيش روي صدور مجوزها و پيگيري در رفع آنها، نظارت و پيگيري مستمر بر فرآيند صدور مجوزها همچنان ضرورت داشته و باید از اولويت هاي سازمان‌ها باشد. **همچنين دانش نيز به عنوان عامل تحول در رونق و جهش توليد** در بخش كشاورزي می‌باشد. با **دانش گستري در محيط كشاورزي** مي توان جهش اساسي در توليدات كشاورزي، كه براي كشاورزي ايران به دليل محدويت هاي اقليمي امكان كمتري دارد، ايجاد نمود**.  توسعه مناسبات جديد در كشاورزي**از جمله**اشتغال دانش مزدمحوري، سهم بري دانش از توليد، مردمي كردن بهره وري كشاورزي** ، ايجاد و راه‌اندازي مراكز خدمات كشاورزي غيردولتي، مراكز مهارت‌آموزي و توان افزايي، انتقال دانش و تكنولوژي هاي نوين كشاورزي، اصلاح نژاد از جمله راهکارهای دیگری در جهت تحقق این امر می‎باشند. حال ما به عنوان مهندس نرم‌افزار باید بتوانیم با ارائه روشهای سیستماتیک به حل این مشکلات کمک کنیم، در واقع با بیان روشن‌تر یک طرح توجیهی ارائه دهیم که بتوان مسیر تولید را هموار کنیم. به عنوان مثال ما با الهام از الگوریتم تقسیم و حل می توانیم مسائل و مشکلاتی که بر سر راه یک واحد تولیدی وجود دارد را ابتدا به مسائل کوچکتر تقسیم کنیم و بعد یکی یکی به حل مسائل کوچکتر بپردازیم و سپس از ترکیب پاسخ‌های بخش‌های کوچکتر به یک راه‌حل کلی دست پیدا کنیم. حال ممکن است که این مسائل کوچکتر خود نیز مستلزم استفاده ازالگوریتم‌های دیگری مثلاً الگوریتم‌های بازگشتی برای حل باشند. همچنین می‌توان از گراف‌ها نیز استفاده کردیا مسئله کوله پشتی برای تولید باکیفیت با کمترین هزینه. اکنون با یک مثال واقعی مراحل، موانع و مشکلات در مسیر تولید یک محصول را بررسی می‌کنیم. مثلاً فرض کنیم در یک کارخانه تولید تیغه اره می‌خواهیم یک تیغه اره تولید کنیم اولین مرحله تهیه ماده اولیه محصول که در اینجا ورق فولادی است. که مراحل خرید آن به صورت زیر است:

ما این فولاد را از خارج از کشوربه مبلغ x تومان می‌خریم و طی فرآیندهایی که روی فولاد انجام می‌شود در نهایت یک محصول با قیمت تمام شده y تومان داریم. از طرف دیگر ما می‌توانیم این محصول نهایی را به جای خرید ماده خام و تولید کامل آن در داخل، یک تیغه نیمه‌آماده با همان قیمت فولاد خام بخریم و در اینجا با انجام فرآیندهای کمتری، محصول نهایی با قیمت تمام شده پایین‌تری تولید کنیم. خوب این اولین مانع بر سر راه تولید داخل است. حال ما باید با بررسی مرحله به مرحله، مشکلات و موانع را کشف و راه‌حلی سیستماتیک جهت حذف این موانع ارائه دهیم. اولین مانع ساختار معیوب تعیین حقوق گمرکی برای ورق خام و تیغه نیمه‌آماده که هر دو دارای یک مبلغ است. مانع بعدی نبود تکنولوژی در داخل جهت فرآوری فولادهای گران‌قیمت(تندبُر). پس ما اگر بتوانیم این موانع را حذف و یا تا حدودی اصلاح کنیم می‌توانیم از توقف تولید جلوگیری کنیم. این مثالی از یک بخش کوچک بود که ما از این قبیل مشکلات بسیار داریم که باید برای تحقق شعار سال در حل آنها بکوشیم.

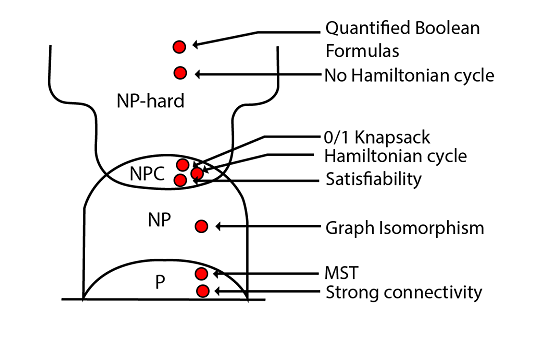
2- ایران جز کشورهایی در دنیاست که به میزان بسیاری زیادی خروجی رشته های مهندسی از دانشگاه های خود دارد که این مهم از یک سوی فرصت و از سویی دیگر تهدید می باشد. در طرف دیگر قضیه، شرایط کشور به نحوی نیست که دولت و نظام بتواند برای همه افراد جامعه از جمله تمامی این مهندسین فارغ‌التحصیل شغل فراهم آورد و از سویی دیگر نیز شرایط اقتصادی کشور و مردم نیز به نحوی نیست که بتوانند سرمایه گذاری‌های عظیمی را در کارها صورت دهند و همچنین این نکته نیز قابل توجه است که کشور در شرایط تحریم به سر می‌برد پس در این شرایط مردم به ویژه جوانان و تحصیل کرده‌ها که بسیاری از آن ها از مدت ها قبل و حتی در دوران دانشجویی ایده‌هایی نیز در ذهن خود پرورانده‌اند، که برای عملیاتی کردن آن ها به یک بستر مناسب نیاز دارند. حمایت همه جانبه  از کارآفرینان و تقویت مادی و معنوی  از آنها می تواند در اشتغال زایی نسل جوان و به تبع آن کاهش سطح آسیب های اجتماعی اثرگذار باشد. همانطور که می‌دانیم الگوریتم روش سیستماتیک حل مسئله است و ما برای حل خیلی از مسائل از الگوریتمهای موجود استفاده می‌کنیم. البته ممکن است برای حل خیلی از مسائل الگوریتم مشخصی وجود نداشته باشد ولی می‌توان با ایده گرفتن از الگوریتم‌های موجود روشها و الگوریتم‌های جدیدی برای حل آنها پیدا کنیم. ما در این مسئله می‌دانیم که یک الگوریتم خاص یا از پیش تعریف شده‌ای برای حل مشکل کارآمدی دانشجویان وجود ندارد بنابراین سعی می‌کنیم از الگوریتم‌های موجود در صورت امکان استفاده کنیم. یکی از الگوریتم‌هایی که استفاده زیادی در زندگی واقعی دارد الگوریتم تقسیم و حل است به این صورت که ما مسائل و مشکلات را به بخش‌های کوچکتر تقسیم کرده و هرکدام از آنها را به صورت جداگانه مورد بررسی قرار می‌دهیم و شاید مجبور شویم هر کدام از این بخشها را مجدد به بخشهای کوچکتر تقسیم کرده و آنها را حل کنیم و در نهایت با جمع بندی جوابهایی که از هر بخش به‌دست می‌آوریم مسئله اصلی را حل کنیم.

خوب به سراغ مسئله می‌رویم؛ برای اینکه یک دانشجو یا مهندس نرم‌افزار موفق و کارامد داشته باشیم باید به یک مرحله عقب تر برگردیم اینکه  عواملی که بعضی دانشجویان را از بقیه موفق‌تر می‌کند چیست؟ یا به بیان دیگر چه مهارت‌هایی باید داشته باشد که تا حدودی موفقیت او تضمین شود؟ **مهارت‌های بنیادی**، **مهارت‌های فراشناختی** و **انگیزه** عواملی هستند که‌تا حد زیادی به موفقیت تحصیلی دانشجویان کمک می‌کنند. مهارت‌های بنیادی شامل فنونِ خواندن می‌شود، این ما را به سمت مهارت بنیادیِ دیگری به نام درک مطلب می‌برد. دانشجویانی که قدرت درک مطلب بالایی دارند، می‌توانند پس از پایان خواندن یک متن، نکته‌ی اصلی هرکدام از ادعاهای مطرح شده، دلیل مطرح شدن آن ادعاها و اینکه هرکدام چه تناقضی با دیگر دیدگاه‌ها دارند را بیان کنند. فراشناخت به این معناست که شما از عملکرد خودتان در کاری که انجام می‌دهید آگاهی کامل داشته باشید و اگر به هدف مورد نظرتان نرسیدید، بدانید اشکال کارتان کجاست. دانشجویان کارآمدتر معمولا دقت بالایی در درک توانایی‌شان از خود بروز می‌دهند و زمانی که در درک یک مطلب ناتوان می‌شوند از مجموعه‌ای از ابزار کمکی و استراتژی‌های مطالعه کمک می‌گیرند تا مشکل‌شان را برطرف کنند.  این ابزار و استراتژی‌ها می‌تواند دوباره خواندنِ مطلب، خلاصه کردن و یادداشت‌برداری از قسمت‌هایی از متن و یا ارتباط برقرار کردن با ایده‌های داخل متن با ایده‌هایی باشد که از قبل دانشجو با آنها آشنایی داشته است. دستورالعمل [ایجاد انگیزه](https://www.chetor.com/35848-%da%86%da%af%d9%88%d9%86%d9%87-%d8%af%d8%b1-%d8%ae%d9%88%d8%af-%d8%a7%d9%86%da%af%db%8c%d8%b2%d9%87-%d8%a7%db%8c%d8%ac%d8%a7%d8%af-%da%a9%d9%86%db%8c%d9%85/) بسیار پیچیده است، همه‌ی دانشجوها انگیزه‌ی لازم برای رسیدن به موفقیت در دانشگاه را ندارند،  اما می‌دانیم که مواد مورد نیازش عبارت است از: 1-ارزش قائل شدن برای خود عملِ مطالعه کردن 2-باور به اینکه اگر برای هدف خاصی مطالعه می‌کنید حتما مهارت آن را داشته و موفق خواهید شد 3- احساس مسئولیت در قبال درست انجام دادن کارها.

پس ما عوامل سه‌گانه برای موفقیت تحصیلی را داریم. دانشجویان، هم باید مهارت‌های بنیادی را برای درک مطالب داشته باشند، هم از [خودآگاهی](https://www.chetor.com/40707-%d8%ae%d9%88%d8%af%d8%a2%da%af%d8%a7%d9%87%db%8c/) بالایی برخوردار باشند و بدانند باید برای آن چه کاری انجام دهند و در آخر برای موفق شدن باید انگیزه‌ی کافی داشته باشند. بعد از بررسی این عوامل و با فرض اینکه این عوامل تا حدودی کارآمدی دانشجویان را تضمین می‌کند حال باید راه‌های ایجاد بستر مناسب در دانشگاه‌ها برای استفاده از دانشجویان کارامد جهت تولید نرم‌افزار با کیفیت و کارا را پیدا کرد. در واقع اگر بخواهیم یک نگاه الگوریتمیک هم به این بخش داشته باشیم باید ببینیم از چه راهکار و روش سیستماتیکی برای گرفتن یک خروجی مناسب از مسئله که همان تولید نرم‌افزاربا کیفیت و دقیق از ورودی مسئله که همان دانشجوی کارامد است استفاده کرد. همانطورکه می‌دانیم دانشگاه ها مستقیما در بخش تولید حضور ندارند، بنابراین با اجرای قراردادهای صنعتی شروع بسیاری از تولیدات از دانشگاه خواهد بود چراکه دانش فنی لازم از سوی جامعه دانشگاهی ایجاد می شود، از طرف دیگر توسعه شرکت‌های دانش بنیان و واحدهای فناور و نوآورکه عمدتا اساتید، دانشجویان و فارغ‌التحصیلان دانشگاهی درآن فعالیت کنند می‌تواند به حل این مسئله کمک کند. یکی دیگر از مواردی که نقش مهمی در تحقق این امر دارند استارت‌اپ‌ها هستند. استارت آپ ها از آنجایی که ایده هایی نو دارند، می توانند در صورتی که درست و با برنامه به پیش بروند، به تولید محصولات خود رسیده و نقش مهمی در تحقق شعار سال هم داشته باشند، لذا توجه و حمایت از استارت آپ ها باید در دستورکار مجموعه های مختلف از جمله دانشگاه‌ها و بخش های علمی ارگان‌های مختلف دولتی و خصوصی باشد. در تعریفی ساده، استارت آپ، کسب و کاری نوپا می باشد که با سرعت بالا، هزینه کم و مهم‌تر از همه داشتن یک ایده نو و جدید شکل می گیرد. یکی از مواردی که دانشجویان بتوانند با فکر و ایده خود، با سرعت بالا و هزینه کم بر روی آن کار کرده و به مرحله ساخت و در صورت سوددهی به تولید انبوه برسانند استارت‌آپ‌ها هستند. حال در این شرایط نیاز است که افراد، گروه ها و ارگان هایی که توانایی و امکان حمایت از استارت آپ ها را دارند، وارد گود شده و به هر طریق ممکن موجبات رشد و پیشرفت این استارت آپ ها را به وجود آورند. در نهایت اگر بخواهیم جمع‌بندی کنیم ما یک مسئله‌‌ای داریم که ورودی ما دانشجوها هستند که در ابتدا باید با تحلیل و بررسی کارآمد بودن آنها را تشخیص داده و در صورت کارا نبودن با دادن راه‌حل به آنها کمک کنیم بعد از اینکه در این قسمت مسئله، خروجی مورد نظر را گرفتیم نوبت به این میرسد که راه‌های ایجاد زمینه کاری مناسب را برای تولید نرم‌افزار باکیفیت تحلیل و بررسی کنیم تا بهترین و معقول‌ترین مسیر را پیدا کنیم تا بتوانیم خروجی کارا و مفید داشته باشیم.

3-P، NP، NP-HARD و NP-COMPLETE کلاسهایی هستندکه مسائل تحت دسته‌بندی آنها قرارمی‌گیرند. قبل از اینکه به تعریف کلاسهای پیچیدگی بپردازیم ابتدا باید ببینیم منظور از زمان چند جمله‌ای چیست؟

زمان چند جمله ای: اگر مطابق با ورودی داده شده در مدت زمان مشخصی مثلاً در عرض یک دقیقه یا ساعت‌ها یک خروجی تولید کنیم. این زمان به چند جمله‌ای معروف است.



**مسائل P (Polynomial)**

مسائل P به مسائلی گفته می شود که الگوریتمی برای آنها وجود دارد که در زمان چند جمله ای قابل حل است. مسئله کوتاهترین مسیر، مسئله درخت پوشای کمینه، مسئله کوله پشتی کسری، مسائل مرتب‌سازی ازاین نوع مسائل هستند. مجموعه ای از مسائل مبتنی بر تصمیم ، در تقسیم بندی مسائل P قرار می گیرند که می توانند در مدت زمان چند جمله ای حل یا تولید شوند. حل مسائل P آسان است .

**مسائل (Non-deterministic Polynomial Problems)NP**

مسائل تصمیم‌گیری هستند که جواب آنها "بله" یا "خیر" است. در واقع بدین معنی است که به صورت غیرقطعی در زمان چند جمله‌ای قابل حل است. مسائل NP در زمان چند جمله ای قابل بررسی هستند ، بدین معنی که با توجه به حل یک مسئله ، می توانیم بررسی کنیم که آیا راه حل در زمان چند جمله ای صحیح است یا خیر. بنابراین به مسائل NP باید خوشبینانه نگاه کرد. در تعریفی مشابه گفته می‌شود که NP مجموعه‌ی مسائل تصمیم‌گیری است که در زمان چندجمله‌ای و با استفاده از ماشین غیرقطعی تورینگ قابل‌حل شدن هستند. مسئله "مرتب بودن" NP است ولی مسئله "مرتب‌سازی" NP نیست چون مسئله تصمیم‌گیری نیست.

مسئله TSP در رده مسائل NP قرار می‌گیرد: Tsp ∈ Np

مسئله Clique هم یک مسئلهNP می‌باشد : " آیا یک خوشه به اندازۀ حداقل k برای G وجود دارد؟"

Clique ∈ Np، چون برای هر گراف x و y مجموعه رأس‌هایC یک جواب.V(x,y) چک می‌کندکه آیاC یک خوشه است

و k |c| ≥.

دو مسئله[**Integer Factorization**](https://en.wikipedia.org/wiki/Integer_factorization)  و[**Graph Isomorphism**](https://en.wikipedia.org/wiki/Graph_isomorphism_problem) نیز مسائل NP هستند.

**مسائلNP-Complete وNp-Hard**

قبل از اینکه به توضیح این دو مسئله بپردازیم ابتدا باید ببینیم Reducibility چیست؟

(Reducibility): اگر بتوانیم یک نمونه از مسئله A را به مسئله B (مسئله NP) تبدیل کنیم ، به این معنی است که A به B قابل تبدیل می‌باشد.

به یک مسئله مانند L یک مسئله‌ی NPکامل(NP-Complete) گفته می‌شود اگر همه مسائل NP را بتوان به آن تبدیل کرد و∈ NP L، به عبارت دیگر مسائل NP-Complete مسائلی هستند که در هردوکلاس NP و NP-Hard وجود دارند، این بدان معناست که مسائل NP-Complete را می توان در زمان چند جمله ای تأیید کرد و هرگونه مسئله NP را می توان در زمان چند جمله ای به این مسائل تبدیل کرد. حال فرض کنیم که R یک مسئله NP-Complete است و به Q قابل تبدیل است، Q حداقل به اندازه R سخت و R یک مسئله NP-Hard است. بنابراین Q حداقل NP-Hard خواهد بود، همچنین ممکن است NP کامل نیز باشد. مسئله صدق‌پذیری یک مسئله NP-Complete است. مسئله SAT، INTEGER PROGRAMMING وVERTEX COVER از جمله مسائل NP-Complete هستند.

در اینجا یک تعریف برای NP-Hard وجود دارد، که X یک مسئله NP-Hard است، اگر یک مسئله NP-Complete به اسم Y وجود داشته باشد، به طوری که Y در زمان چند جمله ای به X قابل تبدیل باشد. مجموعۀ NP-Hard شامل چندهزار مسئله مختلف با کاربردهای فراوان است که تاکنون برای آن‌ها راه حل سریع و قابل انجام در زمان معقول پیدا نشده‌است و به احتمال زیاد در آینده نیز یافت نخواهد شد. این که راه حل سریعی برای آن‌ها وجود ندارد هم اثبات شده‌است. البته ثابت شده‌است که اگر فقط برای یکی از این مسئله‌ها راه‌حل سریعی پیدا شود؛ این راه‌حل موجب حل سریع بقیه مسائل خواهد شد. منظور از راه‌حل سریع آن است که زمان اجرای آن با اندازه ورودی مسئله به صورت چندجمله‌ای رابطه داشته باشد. کلاس NP-hard تنها به مسائل تصمیم‌گیری محدود نمی‌شود. نمونه‌ای از مسائل NP-Hard که درعین‌حال NP-complete نیست «مسئله توقف‌پذیری» است. در این مسئله هدف ساختن الگوریتمی است که مشخص کند آیا یک برنامه کامپیوتری داده شده هیچ‌وقت پایان خواهد یافت (اجزای آن تکمیل خواهد شد و توقف خواهد کرد و یا وارد حلقه نامتناهی خواهد گردید). ورودی چنین الگوریتمی می‌تواند هر برنامه کامپیوتری و طیف داده‌های آن باشد و خروجی چنین الگوریتمی بله یا خیر است. بله یعنی اینکه برنامه ورودی توقف خواهد کرد و خیر یعنی برنامه ورودی در حلقه‌ای نامتناهی گیر خواهد افتاد. الگوریتمی که برای هر برنامه ورودی و طیف داده‌های آن جواب درستی بدهد محاسبه‌ناپذیر است، یعنی هیچ راه‌حل الگوریتمی برای آن وجود ندارد. نتیجه اینکه «مسئله توقف‌پذیری» به کلاس NP تعلق ندارد. از مهمترین مسائل NP-Hard مسئله فروشنده دوره‌گرد و بزرگترین خوشه(maximum clique problem) می‌باشد.