

پاسخنامه آزمون الگوریتم‌های پیشرفته

استاد : دکتر سید علی رضوی ابراهیمی

دانشجو: مریم سادات موردگر- 990190426

تابستان ۱۴۰۰



**سؤالات میان ترم**

**١- در ﻣﻌﻨﻲ و ﻣﻔﻬﻮم ﻋﻠﻤﻲ اﻟﮕﻮرﻳﺘﻢ ﺑﺤﺚ نمایید. درس اﻟﮕﻮرﻳﺘﻢ و الگوریتم‌های موردمطالعه ﺑﺮاى ﻣﻬﻨﺪﺳﻴﻦ نرم‌افزار ﭼﻪ ﻛﺎرﺑﺮد و فایده‌ای دارﻧﺪ؟**

اﻟﮕﻮرﻳﺘﻢ ﻣﺠﻤﻮﻋﻪ ﻣﺘﻨﺎﻫﻲ از دستورالعمل‌هاست ﻛﻪ ﺑﻪ ﺗﺮﺗﻴﺐ ﺧﺎﺻﻲ اﺟﺮا می‌شوند و ﻣﺴﺌﻠﻪ را ﺣﻞ می‌کنند به‌عبارت‌دیگر ﻳﻚ اﻟﮕﻮرﻳﺘﻢ، روش گام‌به‌گام ﺑﺮاى ﺣﻞ ﻣﺴﺌﻠﻪ اﺳﺖ.

اﻟﮕﻮرﻳﺘﻢ روش ﺣﻞ ﺳﻴﺴﺘﻤﺎﺗﻴﻚ ﻣﺴﺌﻠﻪ اﺳﺖ ﻗﺒﻞ از اﻳﻨﻜﻪ ﺣﻞ ﻣﺴﺌﻠﻪ رخ دﻫﺪ ﻣﻬﻢ اﺳﺖ اول ﺑﺪاﻧﻴﻢ ﻛﻪ ﻃﺒﻴﻌﺖ ﻣﺸﻜﻞ ﭼﻴﺴﺖ اﮔﺮ ﻣﺴﺌﻠﻪ را اﺷﺘﺒﺎه درک کرده ﺑﺎﺷید تلاش‌های ﺷﻤﺎ ﺑﺮاى ﺣﻞ ﻣﺴﺌﻠﻪ ﻧﻴﺰ ﻣﻤﻜﻦ اﺳﺖ اﺷﺘﺒﺎه ﻳﺎ ﺑﻴﻬﻮده ﺑﺎﺷﺪ. الگوریتم‌ها داراى ﺧﺼﻮﺻﻴﺎتی هستند مانند: داراى ﻧﻘﻄﻪ ﺷﺮوع و ﭘﺎﻳﺎن هستند ورودى و خروجی‌ها ﻣﺸﺨﺺ ﺑﺎﺷﺪ و ﻣﺮاﺣﻞ ﻛﺎر ﺷﻔﺎف و ﺑﺪون اﺑﻬﺎم ﺗﻌﻴﻴﻦ ﺷﻮد، تکرارپذیرند. ازلحاظ ﻋﻤلی ﻫﻢ بهره‌وری داشته ﺑﺎﺷﻨﺪ و ﺑﺘﻮان ﺑﺎ ﻫﺰﻳﻨﻪ و زﻣﺎن ﻣﻘﻮل ﻳﻌﻨﻲ ﺳﺮﻋﺖ ﺑﻴﺸﺘﺮ و زﻣﺎن ﻛﻤﺘﺮ، ﻣﺴﺌﻠﻪ را ﺣﻞ ﻛﺮد.

ﺗﻔﺎوت ﻣﺘﺪ و اﻟﮕﻮرﻳﺘﻢ: ﻣﺘﺪ ﺷﻴﻮه ﻳﺎ روش ﻧﮕﺎه ﺑﻪ ﻣﺴﺌﻠﻪ اﺳﺖ در ﻣﺘﺪ ﻛﻠﻴﺎت ﻳﺎ ﺟﻬﺖ ﺣﻞ ﻣﺴﺌﻠﻪ ﻣﻌﻴﻦ می‌شود و ﺟﺰﻳﻴﺎت و ﻣﺮاﺣﻞ ﺑﻴﺎن نمی‌شود در اﺻﻞ ﻳﻚ ﻧﺴﺨﻪ عمومی‌تر و کلی‌تر از اﻟﮕﻮرﻳﺘﻢ اﺳﺖ و در آن اﺑﻬﺎم وﺟﻮد دارد وﻟﻲ اﻟﮕﻮرﻳﺘﻢ گام‌های ﺣﻞ ﻣﺴﺌﻠﻪ به‌طور ﺳﻴﺴﺘﻤﺎﺗﻴﻚ و دقیق‌تر اﺳﺖ و در آن ﻫﻴﭻ اﺑﻬﺎﻣﻲ 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. اﻳﻦ روش ﺑﺮاى ﻧﺨﺴﺘﻴﻦ ﺑﺎر ﺗﻮﺳﻂ ﺧﻮارزﻣﻲ موردتوجه ﻗﺮار ﮔﺮﻓﺖ و ﻧﺎم اﻟﮕﻮرﻳﺘﻢ ﻧﻴﺰ ﺑﻪ اﺣﺘﺮام اﻳﻦ داﻧﺸﻤﻨﺪ و از ﻛﻠﻤﻪ اﻟﺨﻮارزﻣﻲ گرفته‌شده 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 به‌راحتی حل کرد.  به‌عنوان‌مثال الگوریتم یک موتور جستجو را در نظر بگیرید. الگوریتم موتور جستجو گوگل به‌طور ساده‌این‌گونه ست که عبارت تایپ‌شده شمارا دریافت کرده و آن را در پایگاه داده‌های خود جستجو می‌کند.

اﻟﮕﻮرﻳﺘﻢ در اﺻﻞ ﻣﻔﻬﻮﻣﻲ اﺳﺖ ﻛﻪ ﻣﺴﺎﺋﻞ را ﺑﺎ اﺳﺘﻔﺎده از دستورالعمل‌های ﭘﺸﺖ ﺳﺮﻫﻢ و ﺑﻪ ﻛﻤﻚ تحلیل‌های رﻳﺎﺿﻲ و ﻣﻨﻄﻘﻲ موردبررسی ﻗﺮار داده و راه‌حل ﻣﻨﺎﺳﺒﻲ ﺑﺮاى آن اراﺋﻪ می‌کند، می‌توان اﻟﮕﻮرﻳﺘﻢ را رویه‌ای ﺗﻌﺮﻳﻒ ﻛﺮد ﻛﻪ ﺷﺎﻣﻞ ﺗﻌﺪاد ﻣﺘﻨﺎﻫﻲ از دستورالعمل‌هاست و ﻫﺪف ﺧﺎﺻﻲ را دﻧﺒﺎل می‌کند، ﻳﻜﻲ از مهم‌ترین اﻫﺪاف در ﻋﻠﻢ ﻛﺎﻣﭙﻴﻮﺗﺮ، ﻓﺮاﻳﻨﺪ ﻃﺮاﺣﻲ اﻟﮕﻮرﻳﺘﻢ ﺑﺮاى ﺗﻮﻟﻴﺪ ﺧﺮوﺟﻲ ﺻﺤﻴﺢ درازای ورودى، در ﻣﺤﺪوده زﻣﺎﻧﻲ ﻣﺘﻨﺎﻫﻲ اﺳﺖ، ﻛﺎراﻳﻲ ﻳﻚ اﻟﮕﻮرﻳﺘﻢ ﺑﺎ آیتم‌های ﻣﺨﺘﻠﻔﻲ ﺳﻨﺠﻴﺪه می‌شود وﻟﻲ مهم‌ترین ﺟﻨﺒﻪ ﺳﻨﺠﺶ ﻣﻴﺰان ﻛﺎراﻳﻲ ﻳﻚ اﻟﮕﻮرﻳﺘﻢ، زﻣﺎن اﺟﺮاى آن اﺳﺖ ﻳﻌﻨﻲ ﺷﻤﺎرش ﺗﻌﺪاد زمان‌هایی ﻛﻪ ﻋﻤﻠﻴﺎت اﺳﺎﺳﻲ اﻟﮕﻮرﻳﺘﻢ اجراشده اﺳﺖ. از دﻳﮕﺮ جنبه‌های ﺳﻨﺠﺶ، ﻓﻀﺎى موردنیاز ﺑﺮاى اﺟﺮاى آن اﺳﺖ.

اﺷﺨﺎص در ﻣﺸﺎﻏﻞ ﮔﻮﻧﺎﮔﻮن ﺑﺮاى اﻧﺠﺎم ﻛﺎر روزاﻧﻪ ﺧﻮد از الگوریتم‌های ذﻫﻨﻲ ﻳﺎ ﻣﻜﺘﻮب ﺧﺎﺻﻲ ﺑﺮاى اﻧﺠﺎم ﻛﺎرﻫﺎ ﻳﺎ اﻧﺠﺎم وﻇﺎﻳﻒ ﺑﻬﺮه می‌برند. ﻳﻜﻲ از ﻣﺒﺎﺣﺚ درﺳﻲ رشته‌های ﻣﻬﻨﺪﺳﻲ ﻛﺎﻣﭙﻴﻮﺗﺮ و ﻓﻨﺎورى اﻃﻼﻋﺎت اﺳﺖ، ﻳﻚ ﻣﺒﺤﺚ پایه‌ای ﺑﺮاى ﻓﻌﺎﻟﻴﺖ در حوزه‌های ﻫﻮش ﻣﺼﻨﻮﻋﻲ، ﻳﺎدﮔﻴﺮى ﻣﺎﺷﻴﻦ و ﻋﻠﻢ داده ﺑﻪ ﺷﻤﺎر می‌آید. ﺗﺴﻠﻂ ﺑﺮ مبحث ﺗﺤﻠﻴﻞ و ﻃﺮاﺣﻲ الگوریتم‌ها ﺑﻪ برنامه‌نویس‌ها ﻧﻴﺰ در ﺣﻞ ﺑﻬﺘﺮ ﻣﺴﺎﺋﻞ برنامه‌نویسی و ﻧﻮﺷﺘﻦ ﻛﺪﻫﺎﻳﻲ ﺑﺎ ﻛﺎراﻳﻲ و ﺳﺮﻋﺖ اﺟﺮاى ﺑﺎﻻﺗﺮ و ﻣﺼﺮف ﺣﺎﻓﻈﻪ ﻛﻤﺘﺮ ﻛﻤﻚ قابل‌توجهی می‌کند. ﻳﺎﻓﺘﻦ راه‌حل ﺑﺮاى ﻣﺴﺎﺋﻞ ﮔﻮﻧﺎﮔﻮن و ﺗﻮاﻧﺎﻳﻲ ﺣﻞ ﻣﺴﺎﺋﻞ ﭘﻴﭽﻴﺪه ﻣﻬﻨﺪﺳﻴﻦ ﻗﺎدر ﺧﻮاﻫﻨﺪ ﺑﻮد ﻛﻪ ﺑﺎ اﺳﺘﻔﺎده از ﻣﺒﺎﺣﺚ الگوریتم‌ها و ﺗﻮﺻﻴﻒ راﺳﺘﻲ آزﻣﺎﻳﻲ و راه‌حل ﺑﺮاى ﻣﺴﺎﺋﻞ واﻗﻌﻲ و ﻣﻄﺮح روز ﺑﻜﺎر ﺑﺒﺮﻧﺪ.

در ﻋﻠﻮم راﻳﺎﻧﻪ ﻳﻚ اﻟﮕﻮرﻳﺘﻢ را روال ﻣﺤﺎﺳﺒﺎﺗﻲ خوش‌تعریف می‌دانند ﻛﻪ ﻣﻘﺪار ﻳﺎ مجموعه‌ای از ﻣﻘﺎدﻳﺮ را به‌عنوان ورودى درﻳﺎﻓﺖ ﻛﺮده و ﭘﺲ از ﻃﻲ چند گام ﻣﺤﺎﺳﺒﺎﺗﻲ، ورودى را ﺑﻪ ﺧﺮوﺟﻲ ﺗﺒﺪﻳﻞ می‌کند. به‌جز اﻳﻦ اﻟﮕﻮرﻳﺘﻢ را اﺑﺰارى ﺑﺮاى ﺣﻞ ﻣﺴﺎﺋﻞ ﻣﺤﺎﺳﺒﺎﺗﻲ ﻧﻴﺰ ﺗﻌﺮﻳﻒ کرده‌اند. ﻳﻚ ﺑﺮﻧﺎﻣﻪ رایانه‌ای ﺑﻴﺎن ﻳﻚ ﻳﺎ ﭼﻨﺪ اﻟﮕﻮرﻳﺘﻢ ﺑﺎ یک‌زبان برنامه‌نویسی اﺳﺖ. معمولاً ﺑﺮاى ﺣﻞ ﻳﻚ ﻣﺴﺌﻠﻪ روش‌ها و الگوریتم‌های ﮔﻮﻧﺎﮔﻮﻧﻲ وﺟﻮد دارد، ﻳﻚ اﻟﮕﻮرﻳﺘﻢ ﻣﻤﻜﻦ اﺳﺖ ﻋﻤﻞ موردنظر را ﺑﺎ دﺳﺘﻮرﻫﺎى ﻣﺨﺘﻠﻒ در مدت‌زمان ﻳﺎ ﻛﺎر ﻛﻤﺘﺮ ﻳﺎ ﺑﻴﺸﺘﺮى ﻧﺴﺒﺖ ﺑﻪ اﻟﮕﻮرﻳﺘﻢ دﻳﮕﺮ اﻧﺠﺎم دﻫﺪ. ﺑﻪ ﻫﻤﻴﻦ دﻟﻴﻞ اﻧﺘﺨﺎب اﻟﮕﻮرﻳﺘﻢ ﻣﻨﺎﺳﺐ و ﻛﺎرا اﻫﻤﻴﺖ زﻳﺎدى در ﻣﻮﻓﻖ ﺑﻮدن و ﻛﺎراﻳﻲ ﺑﺮﻧﺎﻣﻪ رایانه‌ای دارد. الگوریتم‌ها به‌عنوان ﻳﻚ ﻋﻠﻢ ﻣﻄﺮح ﻫﺴﺘﻨﺪ و داﻧﺸﻤﻨﺪان آن‌ها را ﻃﺮاﺣﻲ، ﺗﺤﻠﻴﻞ 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](https://fa.wikipedia.org/wiki/%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87_%D8%B1%D8%A7%DB%8C%D8%A7%D9%86%D9%87%E2%80%8C%D8%A7%DB%8C) دارد. الگوریتم‌ها به‌عنوآن‌یک 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 [روش های تقسیم و غلبه](https://fa.wikipedia.org/wiki/%D8%A7%D9%84%DA%AF%D9%88%D8%B1%DB%8C%D8%AA%D9%85_%D8%AA%D9%82%D8%B3%DB%8C%D9%85_%D9%88_%D8%AD%D9%84)، [روش‌های حریصانه](https://fa.wikipedia.org/wiki/%D8%A7%D9%84%DA%AF%D9%88%D8%B1%DB%8C%D8%AA%D9%85_%D8%AD%D8%B1%DB%8C%D8%B5%D8%A7%D9%86%D9%87)، [روش‌های برنامه‌نویسی پویا](https://fa.wikipedia.org/wiki/%D8%A8%D8%B1%D9%86%D8%A7%D9%85%D9%87%E2%80%8C%D9%86%D9%88%DB%8C%D8%B3%DB%8C_%D9%BE%D9%88%DB%8C%D8%A7)، 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 مرکزیِ](https://fa.wikipedia.org/wiki/%D9%88%D8%A7%D8%AD%D8%AF_%D9%BE%D8%B1%D8%AF%D8%A7%D8%B2%D8%B4_%D9%85%D8%B1%DA%A9%D8%B2%DB%8C) رایانه برای اجرای دستورالعمل‌ها و از [حافظه](https://fa.wikipedia.org/wiki/%D8%AD%D8%A7%D9%81%D8%B8%D9%87_%D8%AA%D8%B5%D8%A7%D8%AF%D9%81%DB%8C) 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](https://fa.wikipedia.org/wiki/%D9%BE%DB%8C%DA%86%DB%8C%D8%AF%DA%AF%DB%8C_%D8%B2%D9%85%D8%A7%D9%86%DB%8C) و [پیچیدگی فضایی](https://fa.wikipedia.org/wiki/%D9%BE%DB%8C%DA%86%DB%8C%D8%AF%DA%AF%DB%8C_%D9%81%D8%B6%D8%A7) 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: اشکال‌زدایی و تحلیل مؤخر (اندازه‌گیری کارایی) است. اندازه‌گیری کارایی عبارت است از فرایند اجرای الگوریتم صحیح بر روی‌داده‌های نمونه‌گیری شده برای به دست آوردن زمان و حافظه موردنیاز توسط کامپایلر. زمان اجرای یک الگوریتم به پارامترهای مختلفی بستگی دارد که ازجمله می‌توان به نوع دستورالعمل‌ها (دستورالعمل‌های جمع، ضرب، نوشتن، خواندن، شرطی و…) کامپایلر مورداستفاده، زبان برنامه‌نویسی، سخت‌افزار به‌کاررفته و پارامتری مثل nکه می‌تواند معرف تعداد ورودی‌ها و خروجی‌ها یا هر دو باشد اشاره کرد.

[تحلیل الگوریتم‌ها](https://fa.wikipedia.org/wiki/%D8%AA%D8%AD%D9%84%DB%8C%D9%84_%D8%A7%D9%84%DA%AF%D9%88%D8%B1%DB%8C%D8%AA%D9%85%E2%80%8C%D9%87%D8%A7) رشته‌ای است که به بررسی کارایی الگوریتم‌ها می‌پردازد. تحلیل الگوریتم‌ها یعنی پیش‌بینی منابع موردنیاز برای اجرای یک الگوریتم، همچون: حافظه، پهنای‌ باند ارتباطی، [سخت‌افزار](https://fa.wikipedia.org/wiki/%D8%B3%D8%AE%D8%AA%E2%80%8C%D8%A7%D9%81%D8%B2%D8%A7%D8%B1_%D8%B1%D8%A7%DB%8C%D8%A7%D9%86%D9%87)، و از همه مهم‌تر، زمان. کارایی یا پیچیدگی هر الگوریتم را با تابعی نشان می‌دهند که تعداد مراحل لازم برای اجرای الگوریتم را برحسب طول داده ورودی، یا میزان محل‌های لازم حافظه را برحسب طول داده ورودی نشان می‌دهد.

**٢-ﻧﺤﻮه ﻋﻤﻞ ﻧﻴﻢ پاک‌کننده در شبکه‌های ﻣﺮﺗﺐ ﺳﺎز را ﺷﺮح دﻫﻴﺪ؟**

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](https://fa.wikipedia.org/wiki/%D9%85%D8%B1%D8%AA%D8%A8%E2%80%8C%D8%B3%D8%A7%D8%B2%DB%8C) یک مدل انتزاعی ریاضی شامل شبکه‌ای از سیم‌ها و واحدهای [مقایسه کننده](https://fa.wikipedia.org/wiki/%D9%85%D9%82%D8%A7%DB%8C%D8%B3%D9%87%E2%80%8C%DA%A9%D9%86%D9%86%D8%AF%D9%87) است که برای [مرتب‌سازی رشته‌ای](https://fa.wikipedia.org/wiki/%D9%85%D8%B1%D8%AA%D8%A8%E2%80%8C%D8%B3%D8%A7%D8%B2%DB%8C_%D8%B1%D8%B4%D8%AA%D9%87%E2%80%8C%D8%A7%DB%8C) 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](https://fa.wikipedia.org/wiki/%D8%A7%D9%84%DA%AF%D9%88%D8%B1%DB%8C%D8%AA%D9%85_%D9%85%D8%B1%D8%AA%D8%A8%E2%80%8C%D8%B3%D8%A7%D8%B2%DB%8C) این است که ترتیب مقایسه‌ها بدون در نظر گرفتن نتیجه مقایسه‌های قبلی، از قبل مشخص‌شده 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. ﻳﻚ ﺷﺒﻜﻪ ﻣﺮﺗﺐ ﺳﺎز، ﻳﻚ ﺷﺒﻜﻪ مقایسه‌گر اﺳﺖ ﻛﻪ رشته‌های ورودى را درﻳﺎﻓﺖ ﻛﺮده و آن را به‌صورت مرتب‌شده در ﺧﺮوﺟﻲ ﻧﻤﺎﻳﺶ می‌دهد به‌بیان‌دیگر ﺑﻪ ﻣﺪارى ﻛﻪ n ورودى را درﻳﺎﻓﺖ ﻛﺮده و ﺑﺎ ﻣﺮﺗﺐ ﻛﺮدن آن‌ها ﺗﻌﺪاد n ﺧﺮوﺟﻲ را ﺗﻮﻟﻴﺪ می‌کند ﭘﺎﻳﻪ و اﺳﺎس شبکه‌ها و الگوریتم‌های مرتب‌سازی ﻣﺒﺘﻨﻲ ﺑﺮ ﻣﻘﺎﻳﺴﻪ و ﺟﺎﺑﺠﺎﻳﻲ ﺟﻔﺖ ﻛﻠﻴﺪﻫﺎ اﺳﺖ.

**شبكه مرتب ساز : Bitonic**

دنباله‌ای كه به‌صورت يكنواخت افزايش می‌یابد و سپس كاهش می‌یابد يا می‌توان آن را به‌صورت گردشی تغيير داد تا يكنواخت افزايش و سپس یکنواخت كاهش می‌یابد بنابراين می‌توان گفت با استفاده از مرتب‌سازی بايتونيك می‌توان يك شبكه مقایسه‌ای را به‌صورت صفرها و یک‌ها مرتب كرده و سپس آن را برای هر عدد دلخواه ديگر بكار برد. شبکه‌های مرتب‌سازی به‌دست‌آمده از مقایسه کننده تشکیل‌شده و تأخیری به میزان دارند که n تعداد مواردی است که باید مرتب شوند. توالی مرتب‌شده یک توالی یکنواخت غیر کاهشی یا غیر افزایشی است. توالی بایتونیک یک توالی با برای برخی مقادیر k، که مقادیر آن در محدوده قرار دارند، با شیفت دورانی این توالی است.

**نيم پاک‌کننده:**

در مرتب‌سازی بايتونيك عمل مرتب‌سازی در چندين مرحله انجام می‌شود به هريك از اين مراحل نيم پاک‌کننده گفته می‌شود.در شبكه مقایسه‌ای half-cleaner ورودی بر روی خط i با ورودی خط٢i-n/ مقايسه می‌شود فرض بر اين است كه ورودی‌ها به‌صورت دو آهنگی يكنواخت افزايش و يا كاهشی باشند بعد از پايان مقايسه حداقل یک‌نیمه تميز خواهيم داشت.

عمق هر نيم پاک‌کننده، برابر با ١ خواهد بود. خط i با خط ٢i-n/ مقايسه شده و مقادير کوچک‌تر را در نيمه بالايی قرار می‌دهد و نيمه پايينی بايتونيك است.

درhalf-cleaner حداقل يكی از نصفه‌ها تميز باشد، در تعداد صفر و یک‌ها، يا تعداد یک‌ها و صفرها مساوی است يا تعداد یک‌ها يا تعداد صفرها كمتر است در هر حالت اگر تعداد صفرها بيشتر از نصف باشد، درواقع يك نصفه صفر خواهيم داشت اگر تعداد یک‌ها بيشتر باشد نصفه يك خواهيم داشت و يك نصفه خواهيم داشت كه تركيبی از صفر و يك است.

در هر حالت half-cleaner يك نصفه تميز خواهيم داشت و آن نصفه ديگر حتماً بايتونيك خواهد بود.

خصوصیات

٠

٠

٠

٠

٠

Bi

to

n

ic

clea

n

٠

٠

١

١

١

٠

٠

٠

Bito

n

ic

١

١

١

Bito

n

ic



اﮔﺮ ورودى ﻧﻴﻤﻪ تمیزکننده ﻳﻚ ﺗﻮاﻟﻲ ﺑﺎﻳﺘﻮﻧﻴﻚ از ٠ و ١ ﺑﺎﺷﺪ در اﻳﻦ ﺻﻮرت ﺗﻮاﻟﻲ ﺧﺼﻮﺻﻴﺎت زﻳﺮ را ﺑﺮآورده می‌کند:

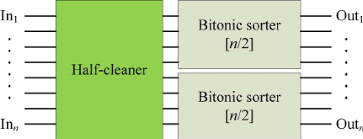
١- ﻫﺮ دونیمه ﺑﺎﻻ و ﭘﺎﻳﻴﻦ ﺑﺎﻳﺘﻮﻧﻴﻚ ﻫﺴﺘﻨﺪ.

٢- هر المان در ﻧﻴﻤﻪ ﺑﺎﻻﻳﻲ حداکثر به‌اندازه ﻫﺮ اِﻟﻤﺎن ﻧﻴﻤﻪ ﭘﺎﻳﻴﻨﻲ ﻛﻮﭼﻚ ﺧﻮاﻫﺪ ﺑﻮد.

٣- ﺣﺪاﻗﻞ ﻧﻴﻤﻲ از آن ﺗﻤﻴﺰ اﺳﺖ(ﻳﺎ ﻫﻤﻪ ﺻﻔﺮ ﻳﺎ ﻳﻚ)

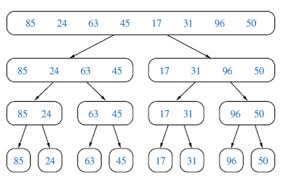
درواقع ﻧﻴﻢ پاک‌کننده ﺑﻴﺎن می‌کند ﻛﻪ ﻫﺮ ﻋﻨﺼﺮ ﺧﺮوﺟﻲ از ﻧﻴﻤﻪ ﺑﺎﻻ ﺣﺪاﻗﻞ به‌اندازه ﻫﺮ ﻋﻨﺼﺮ از ﻧﻴﻤﻪ ﭘﺎﻳﻴﻦ می‌باشد و ﺣﺪاﻗﻞ یک‌نیمه از آن ﺗﻤﻴﺰ می‌باشد.

در شکل زیر یک توالی دو آهنگی یکنواخت افزایشی را نشان می دهد. بعد از مرحله اول مرتب سازی، درسمت چپ ،یک نیم پاک کننده از صفرها و سمت راست، یک توالی دو آهنگی یکنواخت کاهشی تولید خواهد شد. عمق هر نیم پاک کننده، برابر با یک بوده و خواهیم داشت: 1 D(n) =



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 1-n عنصر مرتب، ادغام نماید، حداقل دارای عمق log n خواهد بود. ضمن اینکه ،هرشبکه ادغام صرفنظر از ترتیب ورودی ها به Ω (n log n) مقایسه گر نیاز دارد. بهترین زمان برای مرتب سازی n عنصر در شبکه های مرتب سازی نیز برابر O(lgh2n) 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 SORTER[n] شبکه های مقایسه ای هستند که براساس شبکه های ادغام MERGER[n] مبتنی بر الگوریتم مرتب ساز ادغامی پیاده سازی و در هر مرحله، به صورت بازگشتی 2𝑛 توالی را مرتب می کند. هر 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.

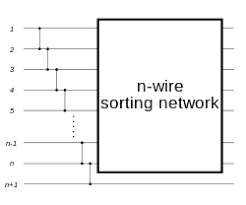


اندازه شبکه های مقایسه ای بوسیله تعداد نهایی مقایسه گرها و تعداد لایه های شبکه به دست میآید. در هریک از این دو پارامتر، پیدا کردن شبکه مرتب سازی بهینه حتی زمانی که شبکه ، به تعداد کمی ورودی محدود می شود، به عنوان یک چالش جدی به شمار میآید.

برخی شبکه های مرتب سازی که از تعداد کم مقایسه گر برخوردار هستند، ازالگوریتم 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، وهمواره یک نیمه مرتب شده است بدین معنی که همه عناصر آن یا صفر است یا یک.



سؤالات پایان‌ترم

**١- ﻧﻘﺶ اﻟﮕﻮرﻳﺘﻢ ﺑﺮاى تأثیرگذاری ﻣﻬﻨﺪﺳﻲ نرم‌افزار در ﺗﺤﻘﻖ ﺷﻌﺎر ﺳﺎل (ﺗﻮﻟﻴﺪ، پشتیبانی‌ها و ﻣﺎﻧﻊ زداﻳﻲ ﻫﺎ) ﭼﻪ می‌تواند ﺑﺎﺷﺪ؟**

به گزارش مشرق،  قریب به ده سال است که مقام معظم رهبری شعارهای ابتدای هرسال 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 هرسال انتخاب شدند.

ﺳﺎل ١٤٠٠ را ﺳﺎل »ﺗﻮﻟﻴﺪ؛ ﭘﺸﺘﻴﺒﺎﻧﻲﻫﺎ، ﻣﺎﻧﻊزداﻳﻲﻫﺎ« نام‌گذاری و تأکید ﻛﺮدﻧﺪ ﻛﻪ »ﺷﻌﺎر اﻧﻘﻼﺑﻲ ﺟﻬﺶ ﺗﻮﻟﻴﺪ« اﻣﺴﺎل ﺑﺎﻳﺪ ﺑﺎ ﺣﻤﺎﻳﺖ همه‌جانبه و رﻓﻊ ﻣﻮاﻧﻊ، ﻛﺎﻣﻞ ﻣﺤﻘﻖ ﺷﻮد. اﻧﺘﺨﺎب ﺷﻌﺎر ﺣﻜﻴﻤﺎﻧﻪ اﻣﺴﺎل ﺗﻮﺳﻂ ﻣﻌﻈﻢ ﻟﻪ ﻧﺸﺎن می‌دهد ﻛﻪ ﻛﺸﻮر ﺑﻴﺶ از ﻫﻤﻴﺸﻪ ﻧﻴﺎزﻣﻨﺪ ﻫﻤﺪﻟﻲ، ﻫﻤﻜﺎرى و ﺗﻼش ﻣﻀﺎﻋﻒ اﺳﺖ ﺗﺎ ﺑﺎ ریل‌گذاری اﻗﺘﺼﺎدى ﺑﻪ ﺳﻤﺖ ﺗﻮﻟﻴﺪ ﺣﺮﻛﺖ ﻛﻨﺪ. در ﻋﺮﺻﻪ ﻓﺮﻫﻨﮓ ﺑﺎ زمینه‌سازی ﺑﺮاى ﭘﺸﺘﻴﺒﺎﻧﻲ از ﻧﻴﺮوﻫﺎى ﺧﻼق و دارﻧﺪﮔﺎن اﻳﺪه و ﻧﻴﺰ ﺑﺎ ﺗﻮﺳﻌﻪ ﻣﺤﺼﻮﻻت ﻓﺮﻫﻨﮕﻲ و ﺧﻼق می‌توان ﺑﻪ رﺷﺪ و ﺗﻮﺳﻌﻪ کسب‌وکارها و اﻗﺘﺼﺎد اﻳﻦ ﺑﺨﺶ دست‌یافت. اﻣﺮى ﻛﻪ در دو ﺳﺎل اﺧﻴﺮ ﺗﻼش ﺷﺪ ﺗﺎ ﺿﺮورت و اﻟﺰاﻣﺎت زﻳﺮﺳﺎﺧﺘﻲ آن در ﺳﻄﺢ دﺳﺘﮕﺎه و دوﻟﺖ ﻃﺮح ﺷﻮد، موردتوجه و ﺑﺮرﺳﻲ ﻗﺮار ﮔﻴﺮد و به‌عنوان ﻣﻌﺎون وزﻳﺮ و ﻣﺴﺌﻮل ﺑﻬﺒﻮد ﻣﺤﻴﻂ کسب‌وکارهاى وزارﺗﺨﺎﻧﻪ ﻧﻴﺰ ﺑﺮ اﻳﻦ ﻓﺮاﻳﻨﺪﻫﺎ ﻧﻈﺎرت دﻗﻴﻖ داﺷﺘﻪ ﺑﺎﺷﻨﺪ چراکه روﻳﻜﺮد ﺗﻮﺳﻌﻪ اﻗﺘﺼﺎد ﻓﺮﻫﻨﮓ و ﺗﻮﺳﻌﻪ ﺻﻨﺎﻳﻊ ﻓﺮﻫﻨﮕﻲ و ﺧﻼق در ﻛﺸﻮر ﻣﺎ ازجمله انگاره‌های ﻧﻮﻳﻦ در ﻣﺒﺎﺣﺚ رﺷﺪ شاخص‌های اﻗﺘﺼﺎدى و ﺗﻮﻟﻴﺪى اﺳﺖ.

یکی از راه‌کارهای مطرح‌شده که شاید شاهد تحقق ﺗﻮﻟﻴﺪ، ﭘﺸﺘﻴﺒﺎﻧﻲﻫﺎ، ﻣﺎﻧﻊ زداﻳﻲﻫﺎ، در پایان سال 1400 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 خام فروشی، 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 ١٤٠٠، ﺳﺎل ﺗﻮﻟﻴﺪ، ﭘﺸﺘﻴﺒﺎﻧﻲﻫﺎ، ﻣﺎﻧﻊ زداﻳﻲﻫﺎ نام‌گذاری ﺷﺪه اﺳﺖ و پرواضح اﺳﺖ ﻛﻪ ﺑﺮاى ﺗﺤﻘﻖ اﻳﻦ ﺷﻌﺎر ﺑﺎﻳﺴﺘﻲ ﺑﻪ ﻛﻠﻴﻪ ﻋﻮاﻣﻞ ﻣﺆﺛﺮ در ﺗﻮﻟﻴﺪ ﺗﻮﺟﻪ ﺷﻮد و ﻓﻘﻂ ﻣﻘﺼﻮد از اﻳﻦ ﺷﻌﺎر، ﭘﺸﺘﻴﺒﺎﻧﻲ از ﻛﺎرﻓﺮﻣﺎﻳﺎن ﻛﻪ ﻋﺎﻣﻞ ﺳﺮﻣﺎﻳﻪ را در ﻣﺠﻤﻮﻋﻪ ﻋﻮاﻣﻞ ﻣﺆﺛﺮ ﺑﺮ ﺗﻮﻟﻴﺪ ﻧﻤﺎﻳﻨﺪﮔﻲ می‌کنند، ﻧﺒﻮده اﺳﺖ و ﺑﺎﻳﺴﺘﻲ ﺑﻪ ﻛﺎرﮔﺮان و ﻛﺎرﻣﻨﺪان ﺷﺎﻏﻞ در بخش‌های ﺗﻮﻟﻴﺪى اﻋﻢ از ﺗﻮﻟﻴﺪ ﻣﺤﺼﻮﻻت و ﺗﻮﻟﻴﺪ ﺧﺪﻣﺎت ﻧﻴﺰ ﺗﻮﺟﻪ و اﻟﺘﻔﺎت ﻻزم ﺻﻮرت ﭘﺬﻳﺮد. در ﺗﺒﻴﻴﻦ ﻋﻮاﻣﻞ ﺗﻮﻟﻴﺪ ﻧﻈﺮﻳﺎت ﻣﺨﺘﻠﻔﻲ وﺟﻮد دارد وﻟﻴﻜﻦ در ﺗﻤﺎم ﺳﻨﺎرﻳﻮﻫﺎى ﻣﺮﺑﻮط ﺑﻪ ﺗﻌﻴﻴﻦ ﻋﻮاﻣﻞ ﻣﺆﺛﺮ در ﺗﻮﻟﻴﺪ، دو ﻋﺎﻣﻞ ﻛﺎر و ﺳﺮﻣﺎﻳﻪ 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 قراردادها، به‌ویژه قراردادهای سنگین [سرمایه‌گذاری](http://www.titrebartar.com/fa/news/166088/%D8%B9%D9%84%D8%AA-%D8%A8%DB%8C-%D8%A7%D8%B9%D8%AA%D9%85%D8%A7%D8%AF%DB%8C-%D9%85%D8%B1%D8%AF%D9%85-%D8%A8%D9%87-%D8%B3%D8%B1%D9%85%D8%A7%DB%8C%D9%87-%DA%AF%D8%B0%D8%A7%D8%B1%DB%8C)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 قراردادها در روز نخست برای پیش‌بینی‌ پذیر کردن فضای کسب‌وکار و به‌عنوان ابزاری برای مدیریت و کاهش ریسک ساخته‌شده‌اند؛ 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](https://www.titrebartar.com/fa/news/50274/%D9%84%D8%B2%D9%88%D9%85-%D8%AD%D8%B1%DA%A9%D8%AA-%D9%81%D8%B6%D8%A7%DB%8C-%DA%A9%D8%B3%D8%A8-%D9%88-%DA%A9%D8%A7%D8%B1-%D8%A8%D9%87-%D8%B3%D9%85%D8%AA)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](https://fa.wikipedia.org/wiki/%D8%A7%D9%84%DA%AF%D9%88%D8%B1%DB%8C%D8%AA%D9%85) 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 انسانیِ دیجیتال مشغول به فعالیت است، آخرین کتاب او با عنوان "فرمول: چگونه الگوریتم‌ها تمامی مسائل ما را حل می‌کنند" می‌باشد.

ﻋﻠﻴﺮﻏﻢ ﺗﻤﺎﻣﻲ پیشرفت‌های ﻓﻨﺎوراﻧﻪ و ﻇﻬﻮر ماشین‌ها ﭘﺲ از اﻧﻘﻼب ﺻﻨﻌﺘﻲ و ﻇﻬﻮر ﻛﺎﻣﭙﻴﻮﺗﺮﻫﺎ و ربات‌ها ﭘﺲ از ﻋﺼﺮ اﻃﻼﻋﺎﺗﻲ و شبکه‌ای و شکل‌گیری ﻓﻀﺎى ﻣﺠﺎزى کسب‌وکار در ﻛﻨﺎر ﻓﻀﺎى واﻗﻌﻲ کسب‌وکار، ﻫﻨﻮز دو ﻋﺎﻣﻞ ﺳﺮﻣﺎﻳﻪ و ﻛﺎر ﺟﺰو ﻋﻮاﻣﻞ ﻣﺆﺛﺮ و ﻏﺎﻟﺐ ﺗﺤﻘﻖ ﺗﻮﻟﻴﺪ ﻣﻄﺮح می‌باشند.

ﻛﺸﻮر اﻛﻨﻮن ﻧﻴﺎزﻣﻨﺪ ﺑﺎزﺧﻮاﻧﻲ ﻗﻮاﻧﻴﻦ، روزآﻣﺪﺳﺎزى ﺳﺎﺧﺘﺎرﻫﺎى دوﻟﺘﻲ و روﻳﻜﺮدﻫﺎى ﻧﻮ ﺷﻮﻧﺪه در اﻗﺘﺼﺎد ﺧﻼق و ﻣﺪﻳﺮﻳﺖ درآﻣﺪﻫﺎى ﻏﻴﺮﻧﻔﺘﻲ اﺳﺖ. در ﭼﻨﻴﻦ وﺿﻌﻴﺘﻲ ﺑﺮداﺷﺘﻦ گام‌های ﻧﻮﻳﻦ ﺑﺮاى ﺗﻮﺳﻌﻪ ﺗﻮﻟﻴﺪات و راه ﻳﺎﻓﺘﻦ ﺑﻪ ﺟﺮﮔﻪ ﻛﺸﻮرﻫﺎى ﺻﺎدرﻛﻨﻨﺪه ﻣﺤﺼﻮﻻت و ﺧﺪﻣﺎت ﻓﺮﻫﻨﮕﻲ ﺑﻪ ﻗﺪرت ﺑﺎﻻ و مهارت ﻣﺪﻳﺮﻳﺘﻲ ﻧﻴﺎز 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 درنتیجه تحقق شعار سال نزدیک‌تر شد.

برای نمونه یک الگوریتم در ادامه نشان داده‌شده است.

خیر

اعلام در حوزه اختيارات دولت نبودن براي حل مشكل

خیر

آیا راه حل

مشکل در حوزه اختیارات دولت است

آیا حل مشکل

از وعده های انتخاباتی

است

بله

خیر

آیا مشکل

به وجود آمده مشکل بزرگی است؟

مشکلات

حل

يتم

الگور

دولت

توسط

مشكل

شروع

شود

بزرگ

مشكل

تا

نيست

مهم

پايان

موجود

مشكل

مورد

در

سخنرانی

ايراد

فراموشی

برعدم

تاكيد

و

پايبندی

ابراز

ها

عده

و

دولت

هيات

جلسات

تشكيل

استفاده

با

مشكل

حل

برلزوم

تاكيد

(

يد

با

)

از

بله

بله

**٢- ﺑﺎ ذﻛﺮ ﻣﺜﺎل و اراﺋﻪ ﻳﻚ اﻟﮕﻮرﻳﺘﻢ اوﻟﻴﻪ (ﻟﺰوﻣﻲ ﺑﻪ اﺛﺒﺎت درﺳﺘﻲ اﻟﮕﻮرﻳﺘﻢ ارائه‌شده وﺟﻮد ﻧﺪارد) ﺗﻼش نمایید ﻗﺪﻣﻲ ﺑﺮاى ﺣﻞ ﻣﺸﻜﻞ ﻛﺎرآﻣﺪى داﻧﺸﺠﻮﻳﺎن ﻣﻬﻨﺪﺳﻲ نرم‌افزار ﺑﺮاى ﻣﺸﺎرﻛﺖ در ﺗﻮﻟﻴﺪ نرم‌افزار در دوران داﻧﺸﺠﻮﻳﻲ و ﻳﺎ پس‌ازآن اراﺋﻪ نمایید؟ (استفاده از الگوریتم‌های مطالعه شده در درس مورد استقبال خواهد بود)**

در طی ۱۰ سال گذشته نقش کلیدی که کاربرد فناوری اطلاعات و ارتباطات می‌تواند در توسعه اجتماعی و اقتصادی کشورها ایفاء نماید، مشخص و قطعی شده است. این امر باعث شده تا میزان توسعه‌یافتگی کشورها از اهمیت برخوردار شده و مدیران هر کشور و طرف‌های تجاری و سرمایه‌گذاران در کشورها مایل باشند تا به‌طور شفاف از وضعیت کاربرد فناوری اطلاعات و ارتباطات مطلع شوند. در پاسخ‌گویی به این نیاز ازیک‌طرف سازمان‌ها و مؤسسات تحقیقاتی روش‌ها و شاخص‌های مختلفی را برای تعیین میزان پیشرفته بودن کاربرد فناوری اطلاعات تعریف نموده با اتکای به آن کشورهای مختلف دنیا را طبقه‌بندی نمایند. از طرف دیگر کشورها خود نیز با جدوجهد در تلاش بوده‌اند تا وضعیت «رقومی» خود را هرچه بیش‌تر شفاف نمایند.

اگرچه در ایران از دیرباز مرکز آمار ایران مسئولیت تهیه آمارهای موردنیاز کشور را بر عهده داشته و در همین ارتباط سرشماری‌ها و مطالعات مختلفی را انجام داده و هرساله 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 پاسخ‌گویی به نیازهای اطلاعاتی موجود نباشد. برای حل این مشکل و به‌روز نگه‌داری دائمی آمارهای موردنیاز برنامه‌ریزان 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 انجام‌شده و بر اساس آن به‌زودی مرکزی به نام مرکز ملی اطلاعات ایجاد خواهد شد.

در طی سال‌های گذشته بخش خدمات و نرم‌افزار در ایران در مقایسه با سخت‌افزار از رشد کم‌تری برخوردار بوده، لیکن در طی دو سال گذشته این روند تغییر کرده و هم‌اکنون نرم‌افزار با رشدی معادل ۴۱% در سال به نسبت از سایر بخش‌ها رشد بیش‌تری دارد. در این بخش ده‌ها شرکت به تولید و عرضه نرم‌افزار، محتوا و خدمات اشتغال دارند. تعداد شرکت‌های نرم‌افزاری فعال در سطح کشور بالغ‌بر ۳۰۰ شرکت است. اغلب شرکت‌ها در این بخش خود اقدام به فروش مستقیم نرم‌افزارهای تولیدی خود می‌نمایند. تعداد شرکت‌های خارجی نرم‌افزاری فعال در بازار نرم‌افزار ایران کم است و آن تعداد کم هم بیش‌تر در زمینه‌های نظیر بانکداری، و بنگاه‌های اقتصادی بزرگ فعالیت دارند. دولت برنامه‌های حمایتی مختلفی را برای صنعت نرم‌افزار تدارک دیده است. ازجمله اعمال جدی قانون منع تکثیر غیرمجاز نرم‌افزارها، حمایت از تولید محصولات جدید و یا ارتقای محصولات موجود در بازار، وجود معافیت‌های مالیاتی برای تولیدکنندگان نرم‌افزار و محتوا، و اعطای وام‌های کم‌بهره را می‌توان ازجمله 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 تدوین‌شده 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 ازجمله 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.  
ازجمله حرکت‌های مهمی که در زمینه نرم‌افزار و خدمات در کشور آغازشده، 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.

13

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.

ﻣﺸﻜﻼت ﺗﻮﻟﻴﺪ نرم‌افزار: اﻓﺰاﻳﺶ ﺳﻄﺢ آﮔﺎﻫﻲ ﻣﺘﺨﺼﺼﺎن ﻋﺮﺻﻪ ﻓﻨﺎورى اﻃﻼﻋﺎت ﺑﺎ ﺗﻜﻨﻮﻟﻮژى روز،اﺳﺘﻔﺎده از قانون‌های ﻣﻨﺴﺠﻢ و یکدست، رﻋﺎﻳﺖ ﻗﺎﻧﻮن کپی‌رایت و در ﻧﻈﺮ ﮔﺮﻓﺘﻦ ﻳﻚ ﻗﺎﻧﻮن ﻣﺘﻤﺮﻛﺰ ﻧﺎﻇﺮ ﺑﺮ ﻛﻠﻴﻪ ﻧﻬﺎدﻫﺎى ﻣﺨﺘﻠﻒ در برنامه‌ریزی و ﻣﺪﻳﺮﻳﺖ، هم‌چنین قانون‌گذاری در زﻣﻴﻨﻪ IT راﻫﻜﺎرﻫﺎﻳﻲ ﻣﻨﺎﺳﺐ ﺑﺮاى ﻛﺎﻫﺶ ﻣﺸﻜﻼت IT در ﻛﺸﻮر به‌حساب می‌آیند. ﺗﻮﻟﻴﺪ نرم‌افزار ﻳﺎ به‌طور ﻣﺤﺪودﺗﺮ ﻣﺘﺪوﻟﻮژى ﺗﻮﻟﻴﺪ نرم‌افزار، ﻣﻨﺤﻨﻲ شناخته‌شده ﺑﺮاى ﻣﺘﺨﺼﺼﻴﻦ و دست‌اندرکاران اﺳﺖ. در اﻳﺮان ﺑﺴﻴﺎرى از اﻓﺮاد در اﻳﻦ زﻣﻴﻨﻪ اﻃﻼﻋﺎت ﻛﺎﻓﻲ دارﻧﺪ و در ﺟﺮﻳﺎن آﺧﺮﻳﻦ دﺳﺘﺎوردﻫﺎ و ﻣﻘﺎﻻت روز ﻫﺴﺘﻨﺪ، اﻣﺎ در پیاده‌سازی سیستم‌های در داﺧﻞ اﻳﺮان، ردﭘﺎى اﻳﻦ صحبت‌ها ﻛﻤﺘﺮ دﻳﺪه می‌شود. ﺑﻪ ﮔﻔﺘﻪ ﻛﺎرﺷﻨﺎﺳﺎن، ﺻﻨﻌﺖ نرم‌افزار در اﻳﺮان ﺑﺎ ﻣﺸﻜﻼت زیرساختی عدیده‌ای روﺑﺮوﺳﺖ، اﻣﺎ تلاش‌های ﺑﺴﻴﺎرى ﺑﺮاى رﻓﻊ اﻳﻦ ﻣﺸﻜﻼت و رﺳﻴﺪن ﺑﻪ اﻳﺪه آل‌ها ﺻﻮرت ﮔﺮﻓﺘﻪ اﺳﺖ. از ﻣﺸﻜﻼت ﭘﻴﺶ روى شرکت‌های تولیدکننده نرم‌افزار در اﻳﺮان می‌توان ﻣﺸﻜﻼﺗﻲ ﻫﻤﭽﻮن ﺑﺎﻻ ﺑﻮدن هزینه‌های ﺗﻮﻟﻴﺪ نرم‌افزار، ﺑﻪ ﻃﻮل اﻧﺠﺎﻣﻴﺪن مدت‌زمان ﺗﻮﻟﻴﺪ، عدم‌حمایت‌های دوﻟﺘﻲ، ﻋﺪم ارﺗﺒﺎط ﺑﺎ تولیدکنندگان ﺧﺎرﺟﻲ، ﺗﻮﻟﻴﺪات ﺳﻔﺎرﺷﻲ، ﺧﺪﻣﺎت و ﭘﺸﺘﻴﺒﺎﻧﻲ ﭘﺲ از ﻓﺮوش، ﻋﺪم رﻋﺎﻳﺖ ﻗﺎﻧﻮن کپی‌رایت، ﻛﻤﺒﻮد ﻧﻴﺮوﻫﺎى ﻣﺘﺨﺼﺺ،ﻧﺒﻮد پشتیبانی‌هاى ﻻزم را ﻧﺎم ﺑﺮد ﻛﻪ ﺑﺎﻋﺚ ﺷﺪه ﺻﻨﻌﺖ نرم‌افزار اﻳﺮان ﻧﺘﻮاﻧﺪ ﺑﻪ ﺟﺎﻳﮕﺎه واﻗﻌﻲ ﺧﻮد دﺳﺖ ﭘﻴﺪا ﻛﻨﺪ.

چالش‌هایی ﻛﻪ ﺑﺮﺧﻲ از آن‌ها ﺑﻪ سیاست‌های ﻛﻼن ﻛﺸﻮر ﻣﺮﺗﺒﻂ و ﺑﺮﺧﻲ دﻳﮕﺮ زاﻳﻴﺪه ویژگی‌های ﺧﺎص نرم‌افزار و دست‌اندرکاران ﺗﻮﻟﻴﺪ و ﺗﻮﺳﻌﻪ آن اﺳﺖ.

یکی از کارشناسان دراین‌ارتباط می‌گوید: تولید نرم‌افزار به دلیل عدم استفاده از ابزار و انرژی فیزیکی قابل‌لمس نیست و ازاین‌رو، 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.

ﺷﻜﺴﺖ در پروژه‌های نرم‌افزارى در ﻫﺮ ﻳﻚ از ﭼﻬﺎر ﻣﻮرد »ﻫﺰﻳﻨﻪ«، »زﻣﺎن«، »ﻛﻴﻔﻴﺖ« و »دﺳﺘﻴﺎﺑﻲ ﺑﻪ اﻫﺪاف« ﻣﻄﺮح می‌شود؛ ﺑﺪﻳﻦ ﻣﻌﻨﺎ ﻛﻪ اﮔﺮ ﭘﺮوژﻫﺎى ﺑﺎ ﺻﺮف ﻫﺰﻳﻨﻪ ﺑﻴﺸﺘﺮ ﻳﺎ زﻣﺎن ﺑﻴﺸﺘﺮ ﻳﺎ باکیفیت پایین‌تر اﻧﺠﺎم ﺷﻮد، ﻋﻠﻴﺮﻏﻢ ﺑﻪ ﭘﺎﻳﺎن رﺳﻴﺪن ﭘﺮوژه، آن را ﺗﻮأم ﺑﺎ ﺷﻜﺴﺖ ﻣﻲداﻧﻴﻢ.

به‌طورکلی دﻻﻳﻞ اﺻﻠﻲ ﺷﻜﺴﺖ پروژه‌های IT در اﻳﺮان را می‌توان ﺑﻪ دودسته ﻋﻮاﻣﻞ داﺧﻠﻲ و ﺧﺎرﺟﻲ ﺗﻘﺴﻴﻢ ﻛﺮد:

ﻋﻮاﻣﻞ داﺧﻠﻲ: ﻣﺪﻳﺮان ﭘﺮوژه کم‌تجربه، ناتوانی‌های شرکت‌های ﺗﻮﻟﻴﺪ نرم‌افزار، ﻗﺮاردادﻫﺎى ﻧﺎﭘﺨﺘﻪ، ﻛﻤﺒﻮد ﻧﻴﺮوى اﻧﺴﺎﻧﻲ ﻣﺘﺨﺼﺺ، مقاومت‌های ﻛﺎرﺑﺮان و سازمان‌ها در ﭘﺬﻳﺮش سیستم‌های ﺟﺪﻳﺪ، ارﺗﺒﺎط ﺑﺎ ﻣﺸﺘﺮﻳﺎن و ﻛﺎرﺑﺮان و ﻋﺪم درﮔﻴﺮ ﻛﺮدن ﻛﺎرﺑﺮان در ﭘﺮوژه می‌باشد.

ﻋﻮاﻣﻞ ﺧﺎرﺟﻲ: ﻧﺒﻮد سرمایه‌گذاری ﻣﻨﺎﺳﺐ ﺑﺮاى ﭘﮋوﻫﺶ و ﺗﺤﻘﻴﻖ در ﺣﻮزه نرم‌افزار، سرمایه‌گذاری ﻛﻢ در ﺑﺨﺶ ﺧﺼﻮﺻﻲ و عدم‌حمایت دوﻟﺖ، ﻋﺪم اﺳﺘﻔﺎده از ﻳﻚ اﺳﺘﺎﻧﺪارد واﺣﺪ، ﻣﺸﻜﻼت ﺣﻀﻮر در ﻣﻨﺎﻗﺼﺎت بین‌المللی، ارزان ﺑﻮدن نرم‌افزار و ﻋﺪم در ﻧﻈﺮ ﮔﺮﻓﺘﻦ آن به‌صورت ﻳﻚ ﻛﺎﻻ، ﺗﻌﻄﻴﻼت رﺳﻤﻲ و برنامه‌ریزی زﻣﺎﻧﻲ، ﻋﺪم ﺷﻨﺎﺳﺎﻳﻲ ﺣﻘﻮق ﻣﻮﻟﻔﻴﻦ 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.

ﻓﺮآﻳﻨﺪﻫﺎى ﻣﺪﻳﺮﻳﺖ ﭘﺮوژه ﺑﺎ ٩ ﺗﻮاﻧﻤﻨﺪى ﺗﻌﺮﻳﻒ می‌شود ﻛﻪ اﻳﻦ توانمندی‌ها ﻋﺒﺎرﺗﻨﺪ از: ﻣﺪﻳﺮﻳﺖ ﻳﻜﭙﺎرﭼﮕﻲ ﭘﺮوژه، ﻣﺤﺪوده، زﻣﺎن، ﻫﺰﻳﻨﻪ، ﻛﻴﻔﻴﺖ، ﻣﻨﺎﺑﻊ اﻧﺴﺎﻧﻲ، ارﺗﺒﺎﻃﺎت، رﻳﺴﻚ و برون‌سپاری و ازآنجایی‌که ﻧﻘﺶ ﻫﺮ ﻳﻚ از ﻋﻮاﻣﻞ ﻓﻮق در ﺗﻮﻟﻴﺪ ﻳﻚ ﻣﺤﺼﻮل نرم‌افزارى ﻛﻠﻴﺪى اﺳﺖ ﺗﻮاﻧﻤﻨﺪى ﻳﻚ ﻣﺪﻳﺮ ﭘﺮوژه در ﺗﻮﻟﻴﺪ نرم‌افزار ﻳﻜﻲ از ﻋﻮاﻣﻞ ﻣﻬﻢ و ﺣﻴﺎﺗﻲ در ﻣﻮﻓﻘﻴﺖ ﭘﺮوژه ﺑﻪ ﺷﻤﺎر می‌رود.

اﻛﺜﺮ ﻣﺪﻳﺮان ﭘﺮوژه نرم‌افزارى در اﻳﺮان، ﺑﺮﻧﺎﻣﻪ ﻧﻮﻳﺴﺎن قدیمی‌تر و ﻳﺎ قوی‌تر ﮔﺮوه ﻫﺴﺘﻨﺪ و ﻫﻤﻴﻦ ﻳﻚ ﻧﻔﺮ در ﺗﻮﻟﻴﺪ ﻳﻚ نرم‌افزار نقش‌های زﻳﺎدى ازجمله ﺗﺤﻠﻴﻞ، ﻃﺮاﺣﻲ و برنامه‌نویسی و ... را بر عهده دارد در ﺣﺎﻟﻴﻜﻪ ﻓﺮآﻳﻨﺪ ﻣﺪﻳﺮﻳﺖ داﻧﺶ و ﺗﻮاﻧﺎﻳﻲ ﺧﺎص ﺧﻮد را می‌خواهد و توانمندی‌هاى ﺧﺎص ﺧﻮد را می‌طلبد.

ناتوانی‌های شرکت‌های ﺗﻮﻟﻴﺪ نرم‌افزار: دست‌اندرکاران ﺗﻮﻟﻴﺪ نرم‌افزار در ﻛﺸﻮر ﻣﺎ ﺑﻴﺸﺘﺮ شرکت‌های ﻛﻮﭼﻚ نرم‌افزارى ﺑﺎ پشتوانه‌های ﻣﺎﻟﻲ اﻧﺪك ﻫﺴﺘﻨﺪ. اﻳﻦ شرکت‌ها عمدتاً ﺑﻪ دﻟﻴﻞ ﻣﺤﺪودﻳﺖ ﻣﻨﺎﺑﻊ از روﻳﻜﺮد ﺗﻮﺳﻌﻪ ﺗﻜﻨﻮﻟﻮژى به‌منظور آﺷﻨﺎﻳﻲ ﺑﺎ اﺑﺰار و روش‌هاى ﻧﻮﻳﻦ ﺗﻮﻟﻴﺪ نرم‌افزار ﻏﺎﻓﻞ می‌شوند.

ﻗﺮاردادﻫﺎى ﻧﺎﭘﺨﺘﻪ: ﺑﻴﺸﺘﺮﻳﻦ ﻣﺸﺘﺮﻳﺎن ﻣﻬﻢ ﺣﻮزه نرم‌افزار در ﻛﺸﻮر ﻣﺎ را سازمان‌هاى دوﻟﺘﻲ و نیمه‌دولتی ﺗﺸﻜﻴﻞ می‌دهند. ﻗﺮاردادﻫﺎى ﺗﻮﻟﻴﺪ نرم‌افزار، ﻗﺮاردادﻫﺎﻳﻲ ﻧﺎﭘﺨﺘﻪ و یک‌طرفه ﻫﺴﺘﻨﺪ و ﻣﺠﺮﻳﺎن ﭼﻨﻴﻦ پروژه‌هایی، باوجود آﮔﺎﻫﻲ ﻧﺴﺒﺖ ﺑﻪ ضعف‌های آن ﺑﻴﺸﺘﺮ اوﻗﺎت ﺑﻪ دﻻﻳﻞ اﻗﺘﺼﺎدى ﻣﺠﺒﻮر ﺑﻪ ﭘﺬﻳﺮش آن‌ها ﻫﺴﺘﻨﺪ.

تجربه‌های ﻧﺎﻛﺎﻓﻲ ﺗﻴﻢ ﭘﺮوژه: ﭼﺎﻟﺶ دﻳﮕﺮى ﻧﻴﺰ ﭘﻴﺶ روى ﻣﺪﻳﺮان شرکت‌های ﻛﻮﭼﻚ و ﻣﺘﻮﺳﻂ وﺟﻮد دارد، ﺑﺮاى ﺗﻤﺎﻣﻲ ﻣﺪﻳﺮﻋﺎﻣﻼن ﻣﻬﻢ اﺳﺖ ﻛﻪ ﻛﺎرﻣﻨﺪاﻧﻲ ﺗﻴﺰﻫﻮش، ﺧﻼق و ﻣﺴﺘﻌﺪ اﺳﺘﺨﺪام ﻛﻨﻨﺪ. ﺷﺎﻳﺪ ﻳﻜﻲ از مهم‌ترین ﻋﻮاﻣﻞ ﻣﻮﻓﻘﻴﺖ پروژه‌های نرم‌افزارى را اﻓﺮاد ﻧﺎم ﺑﺮد. متأسفانه در ﻛﺸﻮر ﻣﺎ ﻛﻤﺒﻮد ﻧﻴﺮوى ﻣﺘﺨﺼﺺ ﻳﻜﻲ از چالش‌های ﻋﻤﺪه اﺳﺖ. عمدتاً می‌توان ﻋﺪم ﺑﻪروز ﺑﻮدن و دﺳﺘﺮﺳﻲ ﺑﻪ ﺗﻜﻨﻮﻟﻮژى روز و اﻳﻨﻜﻪ ﻧﻴﺮوى ﻛﺎر روى ﺗﻜﻨﻮﻟﻮژى ﻗﺪﻳﻤﻲ ﺑﺎ برنامه‌های آﻣﻮزﺷﻲ ﻧﺎﻣﻨﺎﺳﺐ و ﻧﺎﻛﺎﻓﻲ ﭘﻴﻮﻧﺪ ﺧﻮرده اﺳﺖ را ﻳﻜﻲ از ﻋﻮاﻣﻞ ﺿﻌﻒ ﺗﻴﻢ نرم‌افزارى داﻧﺴﺖ.

متأسفانه ﺿﻌﻒ ﻧﻈﺎم آﻣﻮزﺷﻲ در دانشگاه‌ها و اﻳﻨﻜﻪ ﻣﺤﺘﻮى آﻣﻮزﺷﻲ اﻛﺜﺮ دروس داﻧﺸﮕﺎﻫﻲ ﻛﺎرﺑﺮدى و ﻋﻤﻠﻲ ﻧﺒﻮده و صرفاً ﺗﺌﻮرى اﺳﺖ و در ﺑﺎزار ﻛﺎر ﻛﺎرﺑﺮدى ﻧﺪارد را می‌توان ﻳﻜﻲ از ﻋﻮاﻣﻞ ﺿﻌﻒ ﻣﺘﺨﺼﺼﻴﻦ اﻳﺮاﻧﻲ ﻧﺎم ﺑﺮد.

مقاومت‌های ﻛﺎرﺑﺮان و سازمان‌ها در ﭘﺬﻳﺮش سیستم‌های ﺟﺪﻳﺪ: بخش‌های IT ﻫﻤﭽﻨﺎن در اﻏﻠﺐ سازمان‌ها و ﻣﺮاﻛﺰ ﺻﻨﻌﺘﻲ، ﺧﺪﻣﺎﺗﻲ و ﺑﺎزرﮔﺎﻧﻲ ﺳﺎزﻣﺎن ﺿﻌﻴﻔﻲ دارﻧﺪ و از داﻧﺶ و ﻓﻨﺎورى روز، در عرصه‌هایی ﻛﻪ ﺗﻜﺎﻣﻞ و ﺗﺤﻮل در آن، ساعت‌به‌ساعت رخ می‌دهد، ﺑﻪ دور ﻫﺴﺘﻨﺪ ﭼﻮن پروژه‌های IT به‌منظور ﺑﺮاورده ﺳﺎﺧﺘﻦ اﻫﺪاف ﻛﻼن ﺳﺎزﻣﺎن آﻏﺎز می‌شوند، نهایتاً ﺑﺎﻳﺪ در ﺳﻄﻮح ﻋﻤﻠﻴﺎﺗﻲ ﭘﺎﺳﺨﮕﻮى ﻧﻴﺎزﻫﺎى ﻛﺎرﺑﺮان ﺑﺎﺷﻨﺪ.

ﻋﺪم سرمایه‌گذاری ﻣﻨﺎﺳﺐ ﺑﺮاى ﭘﮋوﻫﺶ و ﺗﺤﻘﻴﻖ در ﺣﻮزه نرم‌افزار: اﮔﺮﭼﻪ مساعدت‌هایی در اﺧﺘﻴﺎر بخش‌های ﻣﺨﺘﻠﻒ ﺧﺼﻮﺻﻲ و دوﻟﺘﻲ در ﭼﺎرﭼﻮب طرح‌های ﻣﺨﺘﻠﻒ قرارگرفته اﺳﺖ، ﻋﻤﺪه اﻳﻦ سرمایه‌ها اﻏﻠﺐ ﺑﻪ ﺳﻤﺖ بخش‌های سخت‌افزاری ﺳﺮازﻳﺮ و در ﻋﻤﻞ سرمایه‌گذاری در ﻋﺮﺻﻪ ﺗﻮﻟﻴﺪ و به‌تناسب آن ﺗﺤﻘﻴﻖ و ﭘﮋوﻫﺶ ﻛﺎرﺑﺮدى ﺣﻮزه نرم‌افزار ﻛﻪ در زﻣﺮه سودآورترین عرصه‌های اﻗﺘﺼﺎدى در دﻧﻴﺎ ﺷﻨﺎﺧﺘﻪ می‌شود، ﺑﺴﻴﺎر اﻧﺪك 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 زاییده ویژگی‌های خاص نرم‌افزار و دست‌اندرکاران تولید و توسعه آن است.

در ﻛﺸﻮر ﻣﺎ ﺣﻤﺎﻳﺖ ﻛﺎﻓﻲ از ﺑﺨﺶ ﺧﺼﻮﺻﻲ در زﻣﻴﻨﻪ IT ﺑﻪ ﻋﻤﻞ نمی‌آید. ﺑﺮﺧﻲ ﻣﺸﻜﻼت ﻛﻪ اﻳﻦ ﮔﺮوه ﺑﺎ آن روبه‌رو ﻫﺴﺘﻨﺪ ﻋﺒﺎرﺗﻨﺪ از:

روﻧﺪ ﻃﻮﻻﻧﻲ ﭘﻴﺸﺮﻓﺖ ﻛﺎر در ادارات دوﻟﺘﻲ، ﻋﺪم اﺳﺘﻔﺎده از ﻳﻚ اﺳﺘﺎﻧﺪارد واﺣﺪ، ﻳﻜﻲ از ضعف‌های شرکت‌های ﺗﻮﻟﻴﺪﻛﻨﻨﺪه نرم‌افزارى اﻳﻦ اﺳﺖ ﻛﻪ ﺗﻮﻟﻴﺪات آن‌ها از اﺳﺘﺎﻧﺪاردﻫﺎى ﻻزم ﺑﺮﺧﻮردار ﻧﻴﺴﺖ.

شرکت‌های نرم‌افزارى و ﻣﺸﻜﻼت ﺣﻀﻮر در ﻣﻨﺎﻗﺼﺎت بین‌المللی: در ﺣﺎل ﺣﺎﺿﺮ ﻣﺸﻜﻼت اﺟﺮاﻳﻲ ﺑﺴﻴﺎرى ﺑﺮ ﺳﺮ راه ﺣﻀﻮر شرکت‌های نرم‌افزارى ﺟﻬﺖ ﺣﻀﻮر در ﻣﻨﺎﻗﺼﺎت بین‌المللی وﺟﻮد دارد ﻛﻪ ﺑﺨﺸﻲ از آن ﺑﻪ دوﻟﺖ و ﺑﺨﺶ دﻳﮕﺮى از آن ﺑﻪ ﺧﻮد شرکت‌های نرم‌افزارى برمی‌گردد. اﻳﻦ ﻣﺸﻜﻼت ﻋﺒﺎرﺗﻨﺪ از ﻧﺒﻮد اﻣﻜﺎن ﺻﺪور آﺳﺎن ضمانت‌نامه‌های ﻣﺨﺘﻠﻒ ارزى ﻧﻈﻴﺮ ﺷﺮﻛﺖ در ﻣﻨﺎﻗﺼﻪ، پیش‌پرداخت و ﺣﺴﻦ اﻧﺠﺎم ﻛﺎر ﺑﺮاى شرکت‌های نرم‌افزارى، عدم‌حمایت ﻣﻨﺎﺳﺐ بانک‌ها، ﻣﺸﻜﻼت شرکت‌های نرم‌افزارى، ﻋﺪم ﺷﻨﺎﺳﺎﻳﻲ ﺣﻘﻮق ﻣﻮﻟﻔﻴﻦ و قانون کپی‌رایت.

اﺳﺘﺎﻧﺪاردﻫﺎ: اﺳﺘﺎﻧﺪارد ﻛﺮدن و اﺳﺘﺎﻧﺪارد از ﭘﺎﻳﻪﻫﺎى ﻋﻠﻢ و ﻓﻨﺎورى اﺳﺖ ﻛﻪ در ﭘﻴﺸﺮﻓﺖ ﺻﻨﻌﺖ و اﻗﺘﺼﺎد ﻧﻘﺸﻲ ﺑﻪ ﺳﺰا دارد و ﺑﺎﻳﺪ در ﺟﻬﺖ اﻓﺰاﻳﺶ ﺳﻄﺢ کیفیت‌ها ﺗﻼش ﻛﺮد ﺗﺎ ﺑﻪ ﻓﻨﺎورى ﭘﻴﺸﺮﻓﺘﻪ ﺗﻮﻟﻴﺪ ﻫﺮ ﻣﺤﺼﻮل نائل آﻣﺪ.

ﻳﻜﻲ از ﻣﺸﻜﻼت شرکت‌های نرم‌افزار اﻳﺮاﻧﻲ در ﺑﺎزارﻫﺎى ﺟﻬﺎﻧﻲ ﻋﺪم آﺷﻨﺎﻳﻲ ﺑﺎ اﺳﺘﺎﻧﺪارد نرم‌افزار اﺳﺖ، متأسفانه اﺳﺘﺎﻧﺪارد نرم‌افزار ﻫﻨﻮز در اﻳﺮان به‌درستی ﺗﺪوﻳﻦ و ﻳﺎ تعریف‌نشده اﺳﺖ و ﺗﺎﻛﻨﻮن ﻫﻴﭻ اﻗﺪام ﻣﻨﺴﺠﻤﻲ در ﻛﺸﻮر ﻣﺎ ﺑﺮاى ﺗﻌﻴﻴﻦ اﺳﺘﺎﻧﺪارد ﺗﻮﻟﻴﺪ نرم‌افزار ﺻﻮرت ﻧﮕﺮﻓﺘﻪ و ﻫﻴﭻ اﺟﺒﺎرى ﺑﺮاى ارائه ﺷﻨﺎﺳﻨﺎﻣﻪ نرم‌افزارها وﺟﻮد ﻧﺪاﺷﺘﻪ اﺳﺖ. ازآنجاکه ﺗﻨﻮع ﺣﻮزه ﻋﻤﻠﻜﺮد در زﻣﻴﻨﻪ نرم‌افزار ﺑﺴﻴﺎر زﻳﺎد اﺳﺖ، اداﻣﻪ ﺣﻴﺎت اﻳﻦ ﺻﻨﻌﺖ نمی‌تواند بی‌نیاز از وﺟﻮد اﺳﺘﺎﻧﺪاردﻫﺎى ﻣﻨﺴﺠﻢ و یکدست ﺑﺎﺷﺪ. ﺑﺎ ﺗﻮﺟﻪ ﺑﻪ اﻳﻨﻜﻪ ﻫﺮ تولیدکننده نرم‌افزاری ﻳﻚ روش ﺗﻮﻟﻴﺪ ﺧﺎص ﺧﻮد اﺗﺨﺎذ می‌کند و در ﻣﺮاﺣﻞ ﺳﺎﺧﺖ از ﻫﺎى ﻣﺘﻔﺎوت اﺳﺘﻔﺎده می‌کند، ﻧﺘﻴﺠﻪ ﻛﺎر ﺷﺮﻛﺖﻫﺎى ﻣﺨﺘﻠﻒ ﺑﻪ ﻟﺤﺎظ روﻧﺪ ﻛﻴﻔﻲ و اﺟﺮاﻳﻲ ﺷﻜﻞ و ﺷﻤﺎﻳﻠﻲ ﮔﻮﻧﺎﮔﻮن ﺧﻮاﻫﺪ داﺷﺖ و اﻳﻦ اﻣﺮ اصول ﻳﻜﭙﺎرﭼﮕﻲ در اﻳﻦ ﺑﺎزار را ﻓﺪاى ﺧﻮد ﺧﻮاﻫﺪ ﻛﺮد.

اﻳﻨﺠﺎﺳﺖ ﻛﻪ ﻟﺰوم وﺟﻮد اﺳﺘﺎﻧﺪارد نرم‌افزار به‌وضوح ﺣﺲ می‌شود. ﻛﺎرﺷﻨﺎﺳﺎن نرم‌افزار ﻣﻌﺘﻘﺪﻧﺪ ﻛﻪ ﭘﺸﺘﻴﺒﺎﻧﻲ ﻗﺪرﺗﻤﻨﺪ ﻣﺤﺼﻮﻻت و ﻫﻤﭽﻨﻴﻦ ﻣﺴﺘﻨﺪﺳﺎزى و ﺗﺨﻤﻴﻦ ﻛﻴﻔﻴﺖ ﺗﻨﻬﺎ باوجود ﻳﻚ اﺳﺘﺎﻧﺪارد ﻣﻨﻈﻢ ﻣﻘﺪور اﺳﺖ، وﻗﺘﻲ اﺳﺘﺎﻧﺪارد ﺻﺤﻴﺢ وﺟﻮد داﺷﺘﻪ ﺑﺎﺷﺪ ﻛﺎر ﮔﺮوﻫﻲ ﺑﻴﺸﺘﺮ ﻣﻌﻨﺎ ﭘﻴﺪا می‌کند. هم‌اکنون اﺳﺘﺎﻧﺪاردﻫﺎى ﺟﻬﺎﻧﻲ زﻳﺎدى ﻣﺎﻧﻨﺪIEEE وﺟﻮد ﻛﻴﻔﻴﺖ و ﭘﺸﺘﻴﺒﺎﻧﻲ ﻣﻨﺎﺳﺐ در ﺻﻨﻌﺖ نرم‌افزار دﻧﻴﺎ را ﺗﻀﻤﻴﻦ می‌کنند.

ﻣﺘﺄﺳﻔﺎﻧﻪ ﺑﻪ ﻋﻠﺖ ﻓﻘﺪان اﺳﺘﺎﻧﺪارد ﺑﺮاى ﻣﺴﺘﻨﺪﺳﺎزى ﻣﺤﺼﻮﻻت نرم‌افزارى و بی‌توجهی ﺳﺎزﻧﺪﮔﺎن نرم‌افزار و ﻛﺎرﻓﺮﻣﺎﻳﺎن، نرم‌افزارﻫﺎى تولیدشده در ایران اکثراً ﻓﺎﻗﺪ ﺣﺪاﻗﻞ ﻣﺴﺘﻨﺪات ﻻزم ﻫﺴﺘﻨﺪ.

یکی از ضعف‌های شرکت‌های تولیدکننده نرم‌افزاری این است که تولیدات آن‌ها از استانداردهای لازم برخوردار نیست. در ایران شرکت‌های نرم‌افزاری اغلب در رده شرکت‌های کوچک و با تعداد محدودی نیروی فنی شکل می گیرد و سپس پروژه‌های در دست را تبدیل به یک محصول کرده و به‌صورت عمومی اقدام به فروش آن می‌کنند. که این کار اغلب بدون نیازسنجی در بازار صورت می گیرد، بنابراین محصول تولیدشده از استانداردهای لازم برخوردار نیستند.

ﺑﺮاى ﺗﻮﺳﻌﻪ ﺻﻨﻌﺖ نرم‌افزار ﺑﺎﻳﺪ بر روی ﻓﺮاﻳﻨﺪ اﻧﺮژى زﻳﺎدى ﮔﺬاﺷﺘﻪ ﺷﻮد. ﺗﻮﻟﻴﺪ نرم‌افزار ﺑﺰرگ ﻳﻚ ﻛﺎر ﮔﺮوﻫﻲ اﺳﺖ، ﻳﻚ ﺻﻨﻌﺖ اﺳﺖ اﺟﺮاى اﻳﻦ ﻛﺎر ﺑﺪون ﻓﺮاﻳﻨﺪ ﻣﻤﻜﻦ ﻧﻴﺴﺖ بر اساس آﻣﺎرﻫﺎى ﻣﻮﺟﻮد ﻋﻮاﻣﻞ رﻳﺴﻚ در ﻳﻚ ﭘﺮوژه نرم‌افزارى ﺑﺰرگ و ﺳﻬﻢ ﻫﺮ ﻳﻚ از آن‌ها ﻋﺒﺎرﺗﻨﺪ از ﻧﻴﺮوى اﻧﺴﺎﻧﻲ، ﻓﺮاﻳﻨﺪ ﻛﺎر و ﻓﻨﺎورى و اﺑﺰار روز می‌باشد.

ﻗﺴﻤﺖ ﻋﻤﺪه از ﻋﻮاﻣﻞ رﻳﺴﻚ در ﻧﻈﺮ گرفته‌شده در ﻋﺎﻣﻞ ﻧﻴﺮوى اﻧﺴﺎﻧﻲ ﻣﺎ ﻓﺮاﻳﻨﺪ ﻛﺎر قابل‌کنترل اﺳﺖ. ﺑﻨﺎﺑﺮاﻳﻦ دﻳﺪه می‌شود ﻛﻪ ﻧﻘﺶ ﻓﺮاﻳﻨﺪ ﺗﻮﻟﻴﺪ نرم‌افزار در ﻣﻮﻓﻘﻴﺖ ﻳﻚ ﭘﺮوژه ﺑﺰرگ نرم‌افزارى ﺗﺎ ﭼﻪ ﺣﺪ تعیین‌کننده اﺳﺖ.

ﻧﻜﺘﻪ ﭘﺎﻳﺎﻧﻲ ﻛﻪ از اﻳﻦ اﻃﻼﻋﺎت اﺳﺘﻨﺒﺎط می‌شود اﻳﻦ اﺳﺖ ﻛﻪ ﻋﺎﻣﻞ ﻓﻨﺎورى و اﺑﺰار، درﺻﺪ به‌مراتب پایین‌ترى را ﺑﻪ ﺧﻮد اﺧﺘﺼﺎص داده اﺳﺖ. ﺑﻨﺎﺑﺮاﻳﻦ در آموزش‌ها و برنامه‌ریزی‌های ﻛﺎر ﺑﺎﻳﺪ ﺗﻮﺟﻪ ﺑﻴﺸﺘﺮى ﺑﻪ ﻓﺮاﻳﻨﺪ کارکرد ﺗﺎ ﺑﻪ اﺑﺰار پیاده‌سازی؛ متأسفانه اﻳﻦ اﻣﺮ در ﺑﺴﻴﺎرى از اوﻗﺎت ﻓﺮاﻣﻮش می‌شود.

رعایت توصیه‌های زیر می‌تواند احتمال شکست پرو ه های IT در کشور را کاهش د هد.

* افزایش میزان سطح آگاهی جامعه در مقوله‌ی فناوری اطلاعات آگاهی
* افزایش میزان سطح آگاهی متخصصان عرصه‌ی فناوری اطلاعات با تکنولوژی روز با تا سیس مؤسساتی تحت حمایت دولت برای تربیت و آماده‌سازی نیروی متخصص
* استفاده از استانداردهای منسجم و یکدست
* سعی هر چه بیش‌تر در جهت رعایت قانون کپی‌رایت
* در نظر گرفتن یک‌نهاد متمرکز ناظر بر کلیه‌ی نهاد های مختلف در برنامه‌ریزی و مدیریت و قانون‌گذاری درزمینه‌ی IT

مهندسی نرم‌افزار عبارت است از کاربرد مهندسی برای طراحی نرم‌افزار، توسعه، پیاده‌سازی و نگهداری از نرم‌افزار در یک روش سیستماتیک.

استفاده از الگوریتم دایجسترا: روند الگوریتم دایجسترا مطابق زیر می‌باشد:

انتخاب رأس مبدأ

مجموعه‌ی 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 می گیرد. 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 مجموعه‌ی S اضافه می‌گردد. این کار را دوباره از مرحله‌ی ۴ ادامه داده تا رأس مقصد وارد مجموعه‌ی 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.

**٣- دسته‌های ﺳﺨﺘﻲ ﻣﺴﺎﺋﻞ و ﺗﻔﺎوت آن‌ها را ﺗﺸﺮﻳﺢ نمایید.**

به‌طورکلی مسائل به دودسته حل‌شدنی و حل ناشدنی تقسیم می‌شوند. مسائل حل‌شدنی هم به دو قسمت رام شدنی و رام ناشدنی یا بغرنج تقسیم می‌شوند. برای مسائل حل ناشدنی هیچ الگوریتمی راه‌حلی وجود ندارد. مسائل رام شدنی هم در تئوری و هم در عمل حل‌شدنی‌اند. مسائل رام نشدنی 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 n² گام حل شود. در این مثال می‌گوییم که مسئله از درجه پیچیدگی n² می‌باشد. البته تعداد دقیق گام‌ها بستگی به ماشین و زبان مورداستفاده 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(n² روی یک کامپیوتر نمونه داشته باشد، معمولاً روی اکثر کامپیوتر‌های دیگر نیز پیچیدگی زمانی از مرتبه (O(n² 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(nk) 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 (n2) , O (n3), O (1), O (n log n)

مثال‌های الگوریتم‌های زمان – غیر چندجمله‌ای:

O (2n), O(nn) , 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 مرتبه 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. ازنظر بغرنج بودن یا نبودن، مسائل علوم کامپیوتر سه دسته هستند:

1. مسائلی که الگوریتم‌هایی با پیچیدگی زمانی چندجمله‌ای برای آن‌ها پیداشده است.

2. مسائلی که اثبات‌شده است که بغرنج هستند، یعنی الگوریتم چندجمله‌ای برای آن یافت نمی‌شود.

3. 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 زمانیlogn ، مسئله مرتب‌سازی آرایه‌ها با مرتبه زمانیnlogn، ضرب زنجیره‌ای ماتریس‌ها با مرتبه زمانی n3 دسته اول هستند. همچنین مسائل کوتاه‌ترین مسیر فلوید با مرتبه زمانی n3، درخت دودویی بهینه n3 و مسئله درخت پوشای کمینه پریم n2بااینکه 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 اثبات‌شده 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 قابل‌حل در زمان چندجمله‌ای مترادف هم هستند.

- مسائلی که P نیستند یعنی راه‌حل مرتبه زمانی چندجمله‌ای ندارند، یا رام نشدنی و بغرنج‌اند (Intractable) یا کلاً حل ناپذیرند (Unsolvable). مسائل غیر P فقط برای ورودی‌های کوچک به‌صورت منطقی قابل‌حل‌اند و یا ابداً قابل‌حل 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 NP 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 ازنظر زمان موردنیاز تقسیم‌بندی می‌کنند. به‌طور 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 به‌وسیله 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 nn  یا n! یا 2n جزو چندجمله‌ای‌ها محسوب نمی‌شوند.

به‌طورکلی 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 Nondetermenestic Polynomial ) **‌**:‌ به آن دسته از مسائل تصمیم‌گیری می‌گویند که با داشتن یک جواب این توانایی راداریم که درستی یا نادرستی جواب را با استفاده از یک الگوریتم با پیچیدگی زمانی چندجمله‌ای تعیین کنیم.

|  |  |
| --- | --- |
| تعریف NP  یک مجموعه L ∈ NP است اگر یک الگوریتم A و یک چندجمله‌ای p() وجود داشته باشند | |
| برای همه x ∈ L  یک y وجود داشته باشد با شرط |y| ≤ p(|x|) طوری که A(x,y) = YES در زمان p(|x|) | برای همه x′ ∉ L  یک y′ وجود داشته باشد با شرط |y′| ≤ 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 قابل‌حل 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 قابل‌اجرا 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 قابل‌اجرا 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 آن‌یک الگوریتم معین با مرتبه زمانی چندجمله‌ای نیز پیدا کرد.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | |
| **NPC** |  | NP |  |
|  |  |  |  |
|  | P | | |

رابطه بين 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 قابل‌حل 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 درنتیجه 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 ﻛﻼس NP ﻛﺎﻣﻞ ﻳﺎ 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 به‌کاربرده‌ایم.

به آن الگوریتم از درجه چندجمله‌ای O(nc)  (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 تائید 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 .

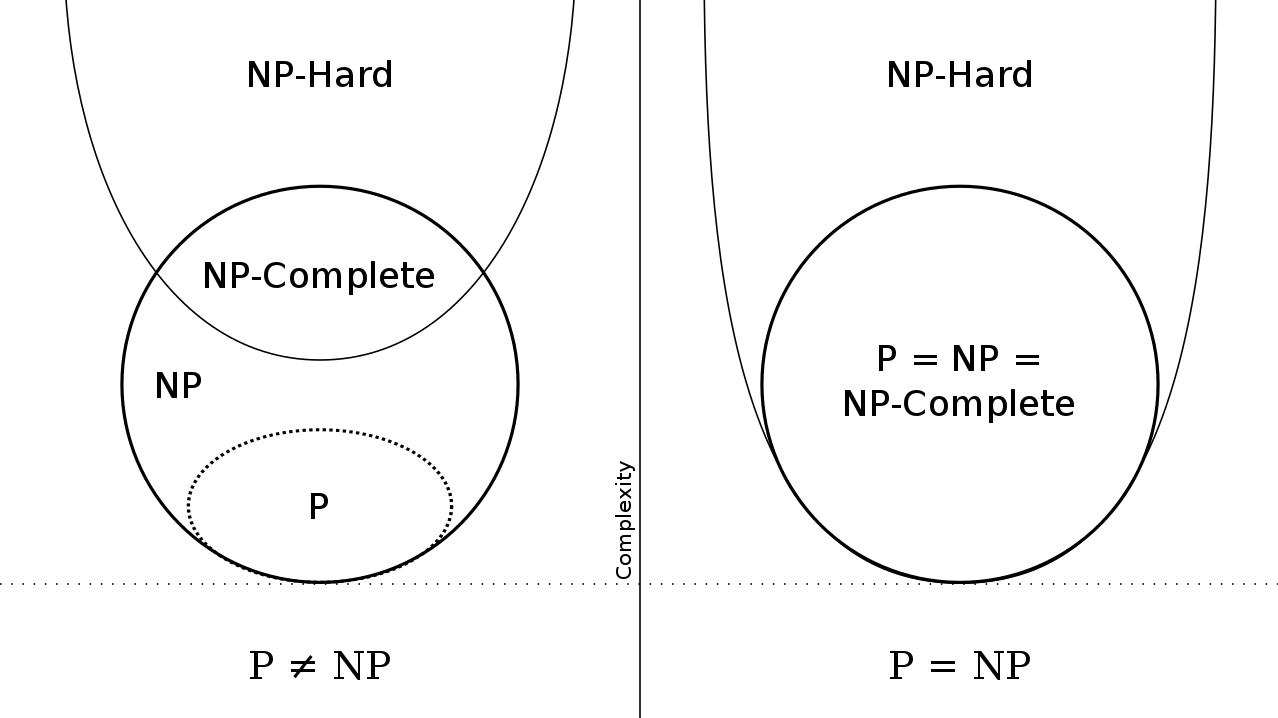
ازجمله مسائل NP کامل یا NP-Complete می‌توان به مسئله کوله‌پشتی یا پیدا کردن دور همیلتونی در گراف یا فروشنده دوره‌گرد اشاره کرد.

به بیان ساده، NP-Completeها مسائلی هستند که اثبات‌شده به‌سرعت قابل‌حل نیستند. در تئوری پیچیدگی NP-Completeها 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 راه‌حل 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 این‌گونه مسائل با شکست مواجه شده‌اند، وقتی‌که مسئله‌ای به‌عنوان NP-Complete معرفی شد، معمولاً این‌طور قلمداد می‌شود که این مسئله در زمان – چندجمله‌ای قابل‌حل شدن نمی‌باشد، یا به بیانی دیگر هیچ الگوریتمی وجود ندارد که این مسئله را در زمان چندجمله‌ای حل نماید. کلاس متشکل از مسائل NP-Compete با نام NP-C نیز خوانده می‌شود.

اگر یک مسئله‌ی بهینه‌سازی ساده باشد مسئله‌ی تصمیم‌گیری مرتبط با آن‌هم 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 آن‌هم 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 به‌صورت 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 درشکل سمت چپ با NP-Complete از آن‌ها یادشده است. همان‌گونه که در شکل پیداست NP هایی بسیار سخت هم هستند که اشتراک آن‌ها با NP کلاس NP کامل را می‌سازد.