

به نام خدا



آزمون درس : الگوریتم پیشرفته

نام و نام خانوادگی دانشجو: فاطمه درویش

شماره دانشجویی: ۹۹۰۱۸۲۱۰۱

زمان آزمون : ۴۸ ساعت

استاد درس: دکتر سیدعلی رضوی

تاریخ: ۱۴۰۰/۰۳/۲۵

تعداد صفحات: ۸

رشته و مقطع تحصیلی : ارشد مهندسی نرم افزار

### سوالات میان ترم

۱- در معنی و مفهوم علمی الگوریتم بحث نمائید. درس الگوریتم های مورد مطالعه برای مهندسين نرم افزار چه کاربرد و فایده ای دارند؟

الگوریتم به معنای روش سیستماتیک حل مسأله است؛ منظور از سیستماتیک این است که نقطه شروع و پایان داشته باشد، مراحل شفاف باشد، تکرارپذیر باشد، امکان پذیر و بهره ور باشد و با زمان و هزینه قابل قبول انجام شود. الگوریتم، ورودی، خروجی، قطعیت و محدودیت دارد.

در این جا سوالاتی مطرح است: برای حل چه نوع مسائلی می توان از الگوریتم استفاده کرد؟ و آیا مسائل با هم متفاوتند؟ در پاسخ باید بگوییم که هر مسأله ای در دنیای واقع، قابل طرح توسط الگوریتم می باشد و از طرفی هر مسأله، می تواند راه حل های مختلفی داشته باشد اما همان طور که ذکر شد منظور، راه حلی است که معقول و عملیاتی باشد. باید توجه داشت که مسائل، شباهت بسیاری به هم دارند؛ می توان مسائل را دسته بندی کرد، آن ها را به مسائل کوچک تر (جزء) تقسیم نمود و اجزاء را انتزاع و مدلسازی کرد، با حل اجزاء مسأله و سرهم بندی آن، مسأله حل خواهد شد؛ و علاوه بر این از اجزاء حل شده می توان در مسائل مشابه استفاده کرد.

آن چه ما در الگوریتم می خوانیم در اصل مسائل انتزاع شده است؛ مثلاً وقتی از گراف صحبت می کنیم این انتزاعی از شبکه های ترافیک (جاده ای، مخابراتی، شبکه های برق رسانی و آبرسانی، شبکه های اینترنت و...) است. با حل این مسائل انتزاع شده، تمام مسائل مرتبط حل می شود. پس می بینیم که مسائل مطرح شده در الگوریتم، انتزاعی از مسائل واقعی در رشته های مختلف (کامپیوتر، سیاست، اقتصاد، فیزیک، ریاضی و...) است.

۲- نحوه عمل نیم پاک کننده در شبکه های مرتب ساز را شرح دهید.

راهنمایی: در مفهوم و معنی عبارت زیر دقت نمائید:

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.

برای درک بهتر عملکرد نیم‌پاک‌کننده‌ها، ابتدا چند مفهوم را مرور می‌کنیم.

: Zero-One Principle

اگر یک شبکه Sort برای ورودی 0/1 پاسخ دهد برای هر ورودی دیگر هم جواب می‌دهد. در واقع  $(\infty)^n$  حالت به ۴ حالت تبدیل می‌شود.  $(\infty)^n \rightarrow (2)^n \rightarrow (2)^2$

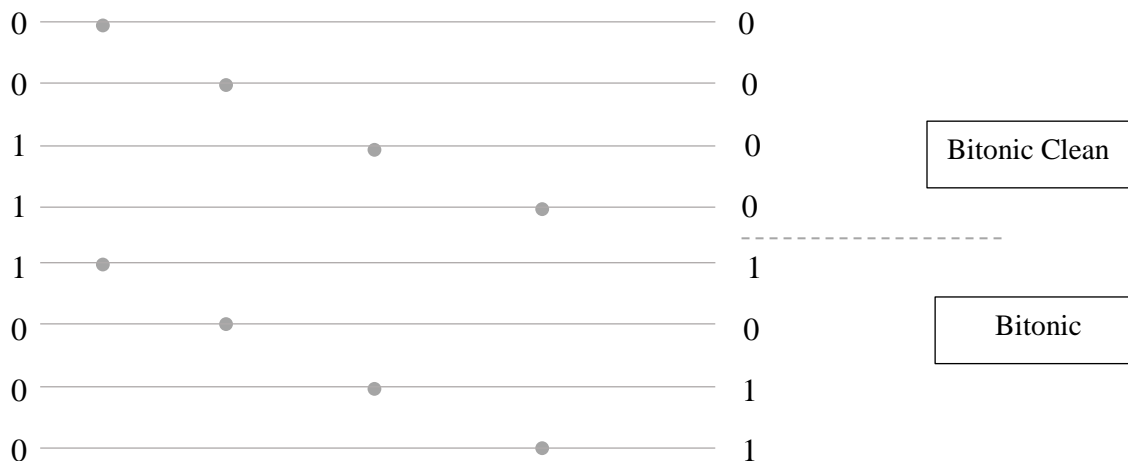
: دنباله‌های Bitonic

دنباله‌ای از اعداد را Bitonic می‌نامیم اگر این دنباله ابتدا صعودی بوده و سپس نزولی باشد و یا به صورت Circularly Shifted (چرخشی/جایگشتی) دارای خاصیت مذکور باشد.

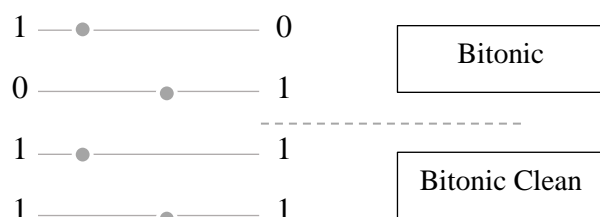
: Half-Cleaner

یک شبکه مقایسه‌گر با  $n$  ورودی و  $n/2$  مقایسه‌گر است که  $n$  معمولاً توانی از 2 است. هر مقایسه‌گر ورودی خط  $i$  را با ورودی خط  $i+n/2$  مقایسه می‌کند. به این ترتیب Half-Cleaner یک دنباله Bitonic را به دنباله‌ای که 50 درصد مرتب است تبدیل می‌کند.

با یک مثال نحوه عمل نیم‌پاک‌کننده را بررسی می‌کنیم. ورودی 00111000 را در نظر بگیرید. قابل توجه است که خط  $i$ ام با خط  $i+n/2$ ام مقایسه می‌شود. خروجی مرحله اول به صورت دو دنباله Bitonic خواهد بود و حداقل یکی از نیمه‌ها حتماً Bitonic Clean است. (اگر تعداد 0ها بیشتر باشد یک نیمه کاملاً 0 و اگر تعداد 1ها بیشتر باشد یک نیمه کاملاً 1 خواهیم داشت).



در مرحله بعد در نیمه Bitonic Clean به دلیل تساوی تمام المان‌ها، در هیچ مقایسه‌ای، جابجایی نخواهیم داشت و نیمه Bitonic به شکل زیر مرتب خواهد شد.



این بار نیز یک نیمه Bitonic Clean داریم که به دلیل بیشتر بودن 1ها شامل 1 است.

می بینیم که با ورودی Bitonic و یک Half-Cleaner، در دو نیمه خروجی، شرایط زیر را خواهیم داشت:

۱. هر دو نیمه Bitonic خواهد بود.
۲. یکی از نیمه ها حتماً Clean است. (0 یا 1)
۳. هر المان در نیمه بالایی حداکثر می تواند به اندازه کوچک ترین المان نیمه پایینی باشد. (المان های نیمه بالایی نمی توانند از المان هایی نیمه پایینی بزرگ تر باشند)

قابل مشاهده است که با یک Half-Cleaner، ورودی Bitonic و Bitonic Sorter یک شبکه مرتب ساز خواهیم داشت.

## سوالات پایان ترم

۱- نقش الگوریتم برای تاثیرگذاری مهندسی نرم افزار در تحقق شعار سال «تولید، پشتیبانی ها و مانع - زداییها» چه می تواند باشد؟

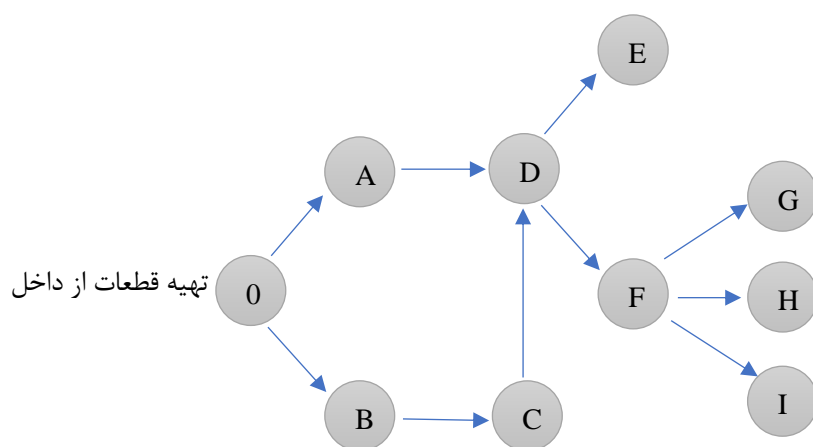
موضوع: طراحی نرم افزاری که داده های تولیدکنندگان در ارتباط با چالش های مطرح شده در ادامه بحث را دریافت کرده و راه حل بهینه را ارائه نماید. به عنوان نمونه استفاده از الگوریتم در صنعت تولید لوازم برقی خانگی را بررسی می نماییم. واضح است که الگوریتم و نرم افزار مورد بحث در زمینه های مختلف صنایع مانند خودرو، کشاورزی، دامپروری، نساجی، فرآورده های چوبی و کاغذی، فرآورده های نفتی، فرآورده های غذایی و ... و خدمات مربوط به آنها قابل تعمیم است.

چالش های تولید:

- تأمین قطعات و مواد اولیه
- پرداخت هزینه های تولید (هزینه مواد خام، هزینه دستمزد، هزینه سربار کارخانه)
- تأمین نیروی متخصص
- معرفی محصول، بازاریابی و فروش

### مثال مورد بررسی: تأمین قطعات و مواد اولیه

یکی از مهم‌ترین چالش‌ها در صنعت تولید لوازم خانگی، تأمین قطعات و مواد اولیه است. قصد داریم یک نرم‌افزار جهت تحلیل روش‌های تأمین مواد اولیه و به دست آوردن روش بهینه داشته باشیم لذا نیاز به الگوریتم داریم. می‌خواهیم از روش عقبگرد (Backtracking) استفاده کنیم. اصل اساسی این روش این است که تعیین کنیم آیا هر راه‌حل پیشنهادی به حل مسأله می‌انجامد یا خیر؟ و به این ترتیب راه‌حل‌های کاندید که منجر به پاسخ نمی‌شود حذف می‌گردد. در مثال مورد نظر ما دو راه جهت تأمین مواد اولیه وجود دارد: تأمین از خارج یا تأمین از منابع داخلی. با توجه به شرایط تحریم‌ها، نرخ ارز و بالا بودن هزینه تمام شده تأمین قطعات از خارج می‌توان آن را بررسی نکرد و به سراغ تأمین از منابع داخلی رفت (البته برای تهیه برخی مواد خام ممکن است مجبور به تأمین از خارج باشیم). برای تأمین مواد اولیه از داخل دو حالت پیش‌روی ما خواهد بود: قطعات آماده خرید وجود دارد یا نیاز به تولید قطعات است. در صورت نیاز به ساخت قطعات موضوع تهیه ماشین‌آلات و فرآیند تولید مطرح می‌شود. تمام این موارد و جزئیات بیشتر در گراف زیر قابل مشاهده است:



شرح رئوس گراف:

A. قطعات آماده خرید

B. ساخت قطعات

C. تهیه ماشین‌آلات و فرآیند تولید

D. مواد خام

E. تهیه مواد خام از داخل

F. تهیه مواد خام از خارج

G. تهیه مواد خام از چین

H. تهیه مواد خام از جنوب آسیا

I. تهیه مواد خام از اروپا

هر کدام از یال‌های بین رئوس، هزینه خاص خود را دارد که با توجه به سوابق اجرایی و تحقیق، قابل محاسبه است.

با این گراف و محاسبه ارزش یال‌ها (مثلاً هزینه تا رأس بعدی) و رئوس (مثلاً کیفیت محصول) و استفاده از الگوریتم‌ها، به دست آوردن بهترین مسیر جهت تأمین قطعات و مواد اولیه انجام شدنی است. در اینجا هدف تأمین قطعه با کمترین هزینه است لذا می‌توانیم از الگوریتم‌های کوتاه‌ترین مسیر مانند دیکسترا (Dijkstra's Algorithm) استفاده کنیم.

یک روش دیگر برای حل مساله مورد نظر (تأمین قطعات و مواد اولیه)، استفاده از روش حل مساله کوله پشتی صفر و یک است. به این ترتیب که سقف هزینه‌ی ممکن برای تأمین قطعات (وزن مجاز کوله پشتی) مشخص شده و زیرمجموعه‌های ممکن از قطعات قابل تأمین (ارزش کلی کوله پشتی) تهیه شود. در نهایت بهترین زیرمجموعه (از نظر اولویت تهیه قطعات و هزینه آن‌ها) قابل انتخاب است.

برای تمام موارد دیگر از جمله پرداخت هزینه‌های تولید (هزینه مواد خام، هزینه دستمزد، هزینه سربار کارخانه)، تأمین نیروی متخصص و بازاریابی و فروش، گراف راه‌حل‌های ممکن و انتخاب بهینه‌ترین راه‌حل قابل پیاده‌سازی است.

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

طبق تعریفی که در پاسخ سوال ۱ نیز داشتیم الگوریتم به معنای روش سیستماتیک حل مساله است. برای نوشتن الگوریتم باید سه عامل اصلی را شناسایی کنیم: ۱- مقادیر معلوم (داده‌ها) ۲- خواسته‌های مساله ۳- عملیات (دستورات و روابط منطقی داده‌ها جهت رسیدن به خواسته‌های مساله)

در موضوع مورد بحث عوامل اشاره شده در بالا از قرار زیر است:

۱. مقادیر معلوم: دانش دانشجوی رشته مهندسی نرم‌افزار

۲. خواسته مسأله: مشارکت مؤثر در تولید نرم افزار در زمان دانشجویی و پس از آن

۳. عملیات: تحلیل و بررسی مشکلات، ایجاد فرآیند و پیش نیازها، بهینه سازی، دوام و گسترش

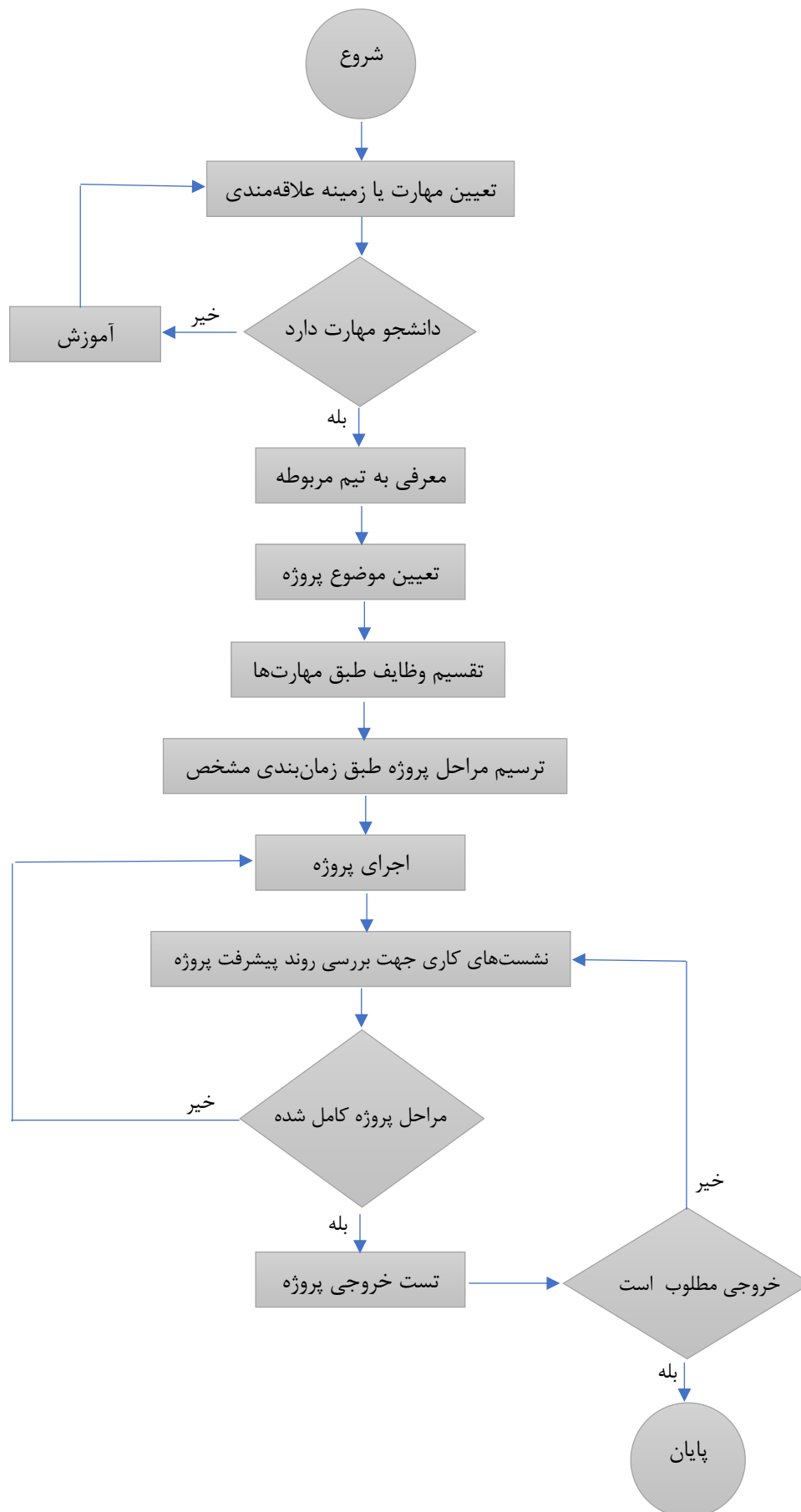
برای ارائه الگوریتم، ابتدا باید مشکل را به خوبی بشناسیم و دلایل ناکارآمدی دانشجویان مهندسی نرم افزار در تولید نرم افزار را بررسی کنیم. از مشکلات این حوزه می توان به این موارد اشاره کرد: ضعف کار تیمی، نداشتن نگاه اقتصادی به پروژه های دانشجویی یا به صرفه نبودن پروژه ها در بازار کار، عدم مستندسازی پروژه ها یا مستندسازی ضعیف آن ها، انجام پروژه ها به صورت سنتی و بدون در نظر گرفتن اصول مهندسی، عدم شناخت و عدم پیش بینی درست ریسک های پروژه که منجر به شکست یا بهره ور نبودن آن می شود، نداشتن تخمین درست از هزینه های انجام یک پروژه، عدم مهارت کافی دانشجویان در کاربرد زبان های برنامه نویسی

جهت تشکیل تیم و تعریف، انجام و تحویل پروژه می توان مراحل دیاگرام زیر را در نظر داشت.

نکات قابل توجه در دیاگرام ارائه شده:

- ✓ در بخش معرفی دانشجویان به تیم های نرم افزاری باید به عواملی چون مهارت افراد، زمینه علاقه مندی و توانایی کار تیمی توجه شود. (تأمین نیروی انسانی مورد نیاز)
- ✓ در تعیین موضوع پروژه لازم است نیازهای بازار کار در نظر گرفته شود تا منافع اقتصادی تیم نرم افزاری دانشجویی تأمین گردد.
- ✓ مستندسازی در تمام بخش های پروژه باید به صورت کامل و مدون انجام گیرد به صورتی که جهت توسعه پروژه یا استفاده در دیگر پروژه های تیم قابل استفاده باشد.

قابل توجه است که در تمام مراحل (تأمین نیرو و تشکیل تیم های نرم افزاری، تعیین موضوع پروژه یا گرفتن سفارش از بازار کار، تخمین هزینه انجام پروژه و...) جهت انجام بهینه کار، همانطور که در پاسخ سوال ۱ پایان ترم نیز بیان شد می توانیم گرافهای درخت، پوشا یا کامل را مطابق با نیاز تشکیل داده و بهترین راه حل را جهت تأمین خواسته برگزینیم.



### ۳- دسته‌های سختی مسایل و تفاوت آن‌ها را تشریح نمائید.

مسائل از نظر درجه سختی، به چهار دسته تقسیم می‌شوند:

۱. P (Polynomial-time algorithm)

۲. NP (Non-deterministic Polynomial time)

۳. NP-Complete

۴. NP-Hard

**مسائل دسته P**، مسائلی هستند که در بدترین حالت الگوریتمی چندجمله‌ای (منظور چندجمله‌ای معقول اغلب با درجه ۲ و ۳ می‌باشد) برای آن‌ها ارائه شده است. به طور کلی با  $O(n)^k$  نمایش داده می‌شود که  $k$  مقداری ثابت است.

**مسائل دسته NP**، مسائلی هستند که به طور غیرقطعی در زمان چندجمله‌ای، قابل حل می‌باشند لیکن هنوز راه‌حلی برای آن‌ها ارائه نشده است. (ثابت نشده که در زمان چندجمله‌ای قابل حل نیستند؛ امیدوارنه می‌گوییم که راه‌حل چندجمله‌ای دارند) قابل توجه است که منظور از NP، Non-Polynomial نیست. برخی مسائل NP نیستند چون اصلاً مسأله تصمیم‌گیری نیستند مثلاً مرتب سازی NP نیست اما تعیین مرتب بودن یا نبودن از نوع NP است.

**مسائل دسته NP-Complete** : بقیه مسائل به آن تبدیل شوند و خودش NP باشد.

**مسائل دسته NP-Hard** : : بقیه مسائل به آن تبدیل شوند و خودش NP نباشد.

سوال مهمی که اینجا مطرح می‌شود این است که ما مسأله غیرچندجمله‌ای (Non-Polynomial) داریم؟ برای پاسخ، این نکته قابل ذکر است که برای هیچ کدام از مسائل مهمی که تا امروز مطرح شده‌است ثابت نشده که پاسخ چندجمله‌ای برای آن‌ها وجود ندارد.