

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

ساختمان‌های داده

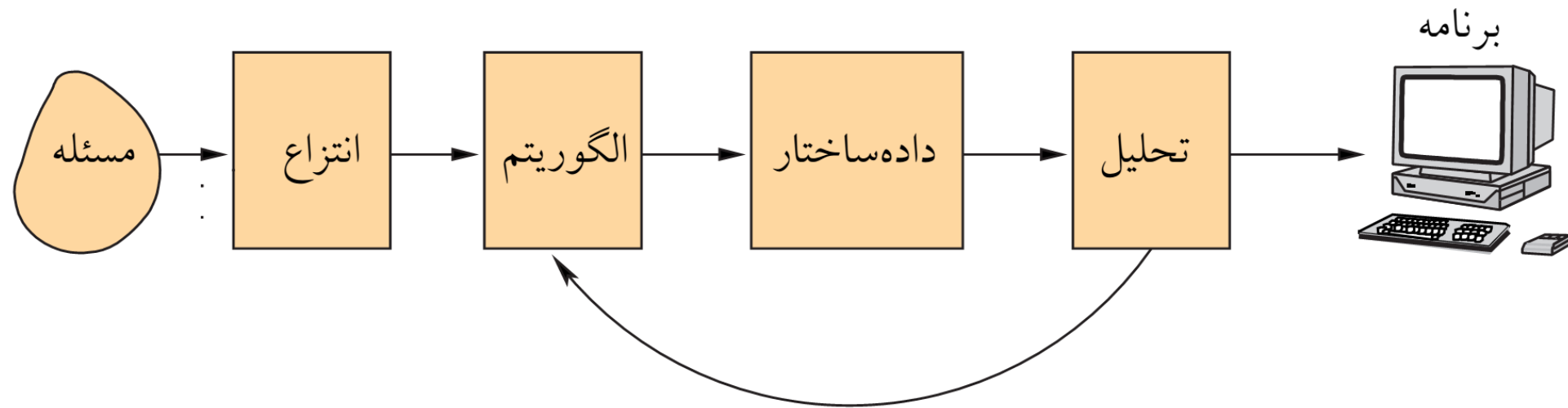
جلسه ۲

مجتبی خلیلی
دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

حضور و غیاب

○ بعد از ترمیم، حضور و غیاب خواهیم داشت.

مراحل حل مسئله



مسئله

○ یک مسئله واقعی حاوی جنبه‌ها و ویژگی‌های مختلف و اغلب متعددی است که به برخی از آنها در طراحی راه حل و پیاده‌سازی چندان نیازی نیست. گام اول در حل هر مسئله شناسایی کامل و تفکیک این ویژگی‌هاست به طوری که پس از پالایش از آن بتوان یک مدل ساده قابل حل ساخت. به گونه پالایش شده مسئله مدل انتزاعی می‌گوییم.

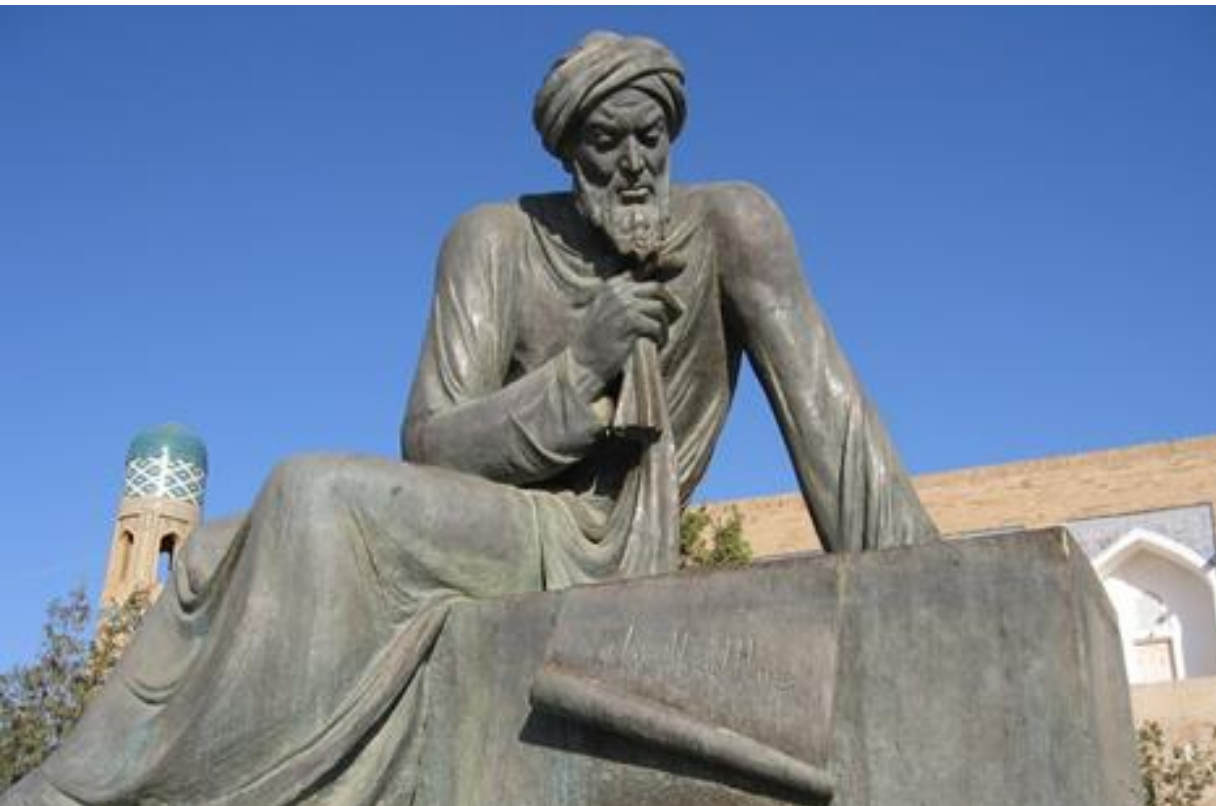
○ مثال

الگوریتم

○ ابوجعفر محمد بن موسی خوارزمی (الخوارزمی) در قرن سوم

○ الجبر و المقابله

○ Algorithm



الگوریتم

○ الگوریتم اقلیدسی برای محاسبه بزرگترین مقسوم علیه مشترک دو عدد نامنفی

gcd (a, b):

```
1 if b==0  
2   return a  
3 else a=b, b=a mod b, goto 1
```

$$\text{gcd}(10,25) = \text{gcd}(25,10) = \text{gcd}(10,5) = \text{gcd}(5,0) = 5$$

الگوریتم

- الگوریتم: مجموعه‌ای از دستورالعمل‌های دقیق که بر روی ورودی اعمال شده و در یک زمان متناهی به خروجی (حل مسئله) منتهی می‌شود.
- مثال: مرتب‌سازی حبابی، جستجوی دودویی

الگوریتم



درستی ○

ساختمان داده / داده ساختار

○ ساختمان داده یا داده ساختار: شیوه‌ای برای ذخیره و سازماندهی داده‌ها به منظور تسهیل در دسترسی و اصلاح/تغییر آنها

- مثال: آرایه، لیست پیوندی، پشته، صف، درخت

ساختمان داده/داده ساختار

○ الگوریتم طراحی شده مشخص می کند که داده های مسئله باید به وسیله چه ساختارهایی عرضه شوند و بر روی هر قلم داده آن چه اعمالی انجام می گیرد. این اطلاعات کمک می کند تا برای ذخیره داده، ساختارهای مناسب انتخاب یا طراحی شوند.

○ ساختمان داده خوب به طراحی الگوریتم خوب کمک می کند.

تحلیل الگوریتم‌ها

○ چه موقع می‌گوییم یک الگوریتم خوب است؟

تحلیل الگوریتم‌ها

○ الگوریتم طراحی شده را با توجه به داده ساختارهای انتخابی می‌توان تحلیل کرد و میزان زمان اجرا و حافظه مصرفی الگوریتم را برای اندازه‌های مختلف ورودی حدس زد.

تحلیل الگوریتم‌ها

○ اگر این الگوریتم پس از تحلیل راضی کننده نبود لازم است مراحل طراحی الگوریتم و انتخاب داده ساختارهای مورد نیاز تکرار شود. این مراحل معمولاً مستقل از یک زبان برنامه نویسی خاص است.

اهداف آموزشی

- یادگیری مبانی ساختمانهای داده و الگوریتمها
 - ساختمان داده‌های کلاسیک: صف، پشته، لیست، درخت و ...
- یادگیری الگوها و فرآیندها برای سازماندهی و پردازش اطلاعات
 - درک درست برای ارزیابی کارایی این الگوها و الگوریتمها
 - درک trade-off بین انتخاب‌های مختلف و گزینش بهترین مورد
- کلاس درباره چگونه برنامه بنویسیم نیست! بلکه هدف طراحی، تحلیل و پیاده‌سازی است.
 - چیزی بین تئوری و مهندسی

اهداف آموزشی

○ از این درس به بعد شما یاد میگیرید مثل یک مهندس کامپیوتر فکر کنید.

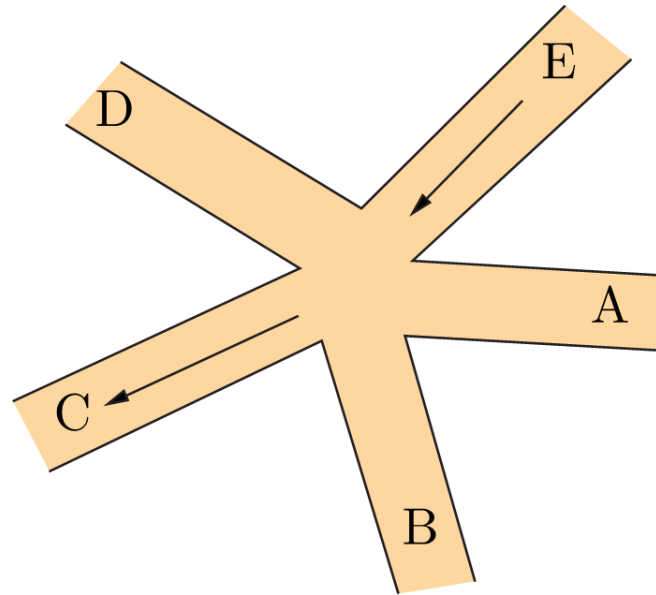
- دیگر نباید همانند کد C++ فکر کنید.
- اینگونه باید فکر کنید: این یک مسئله مرتب سازی است؛ این یک مسئله جستجو است؛ یک مسئله پشته است.
- درک trade-off ها
 - زمان، فضا
 - عملکرد، سادگی

مثال

○ مجموعه دروسی با پیش‌نیازهایشان داده شده است. الگوریتمی بنویسید که طرز صحیح گرفتن این دروس را برگرداند.

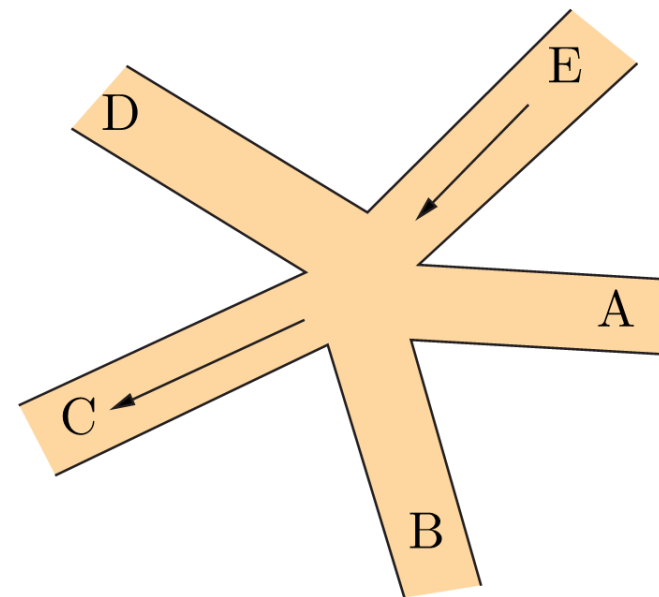
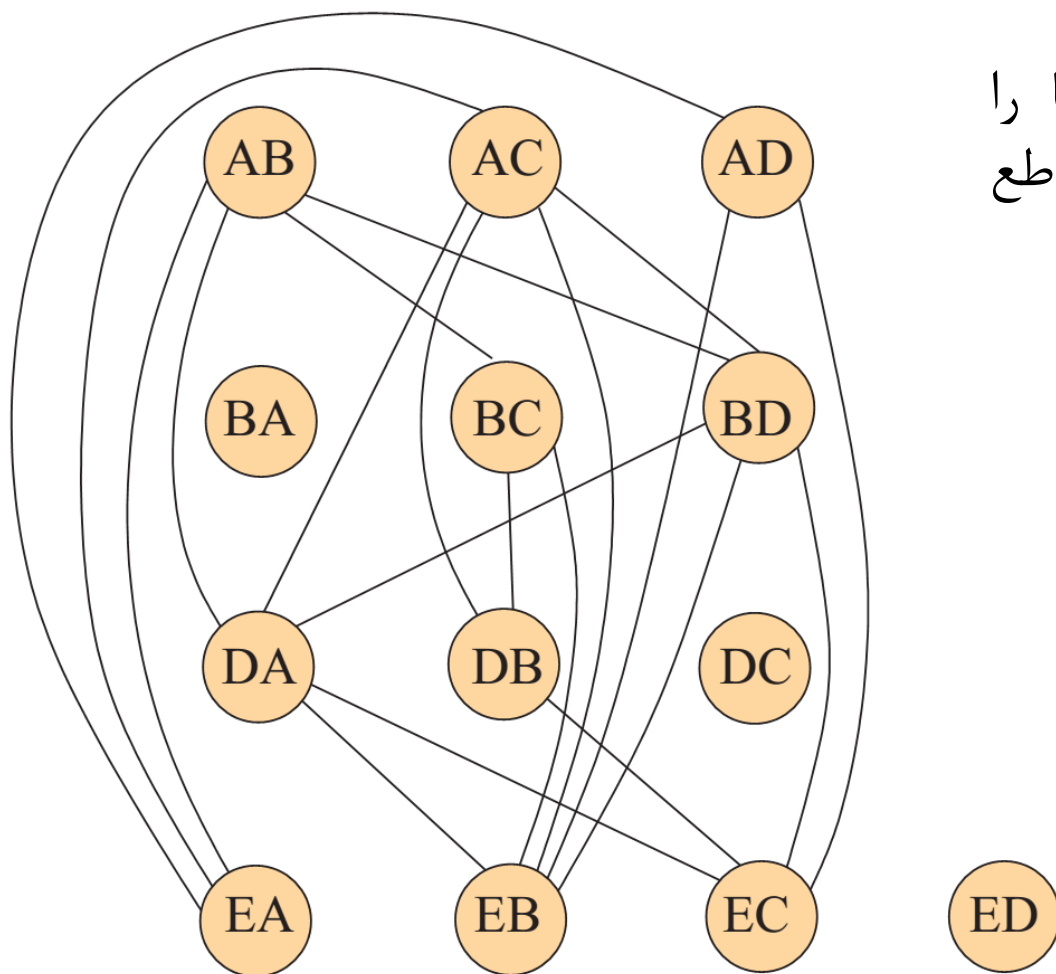
مثال

○ برنامه‌ریزی زمانی برای چراغ‌های راهنمای مربوط به شکل زیر به طوری که تعداد زمان‌های مختلف چراغ راهنمایی کمینه باشد.



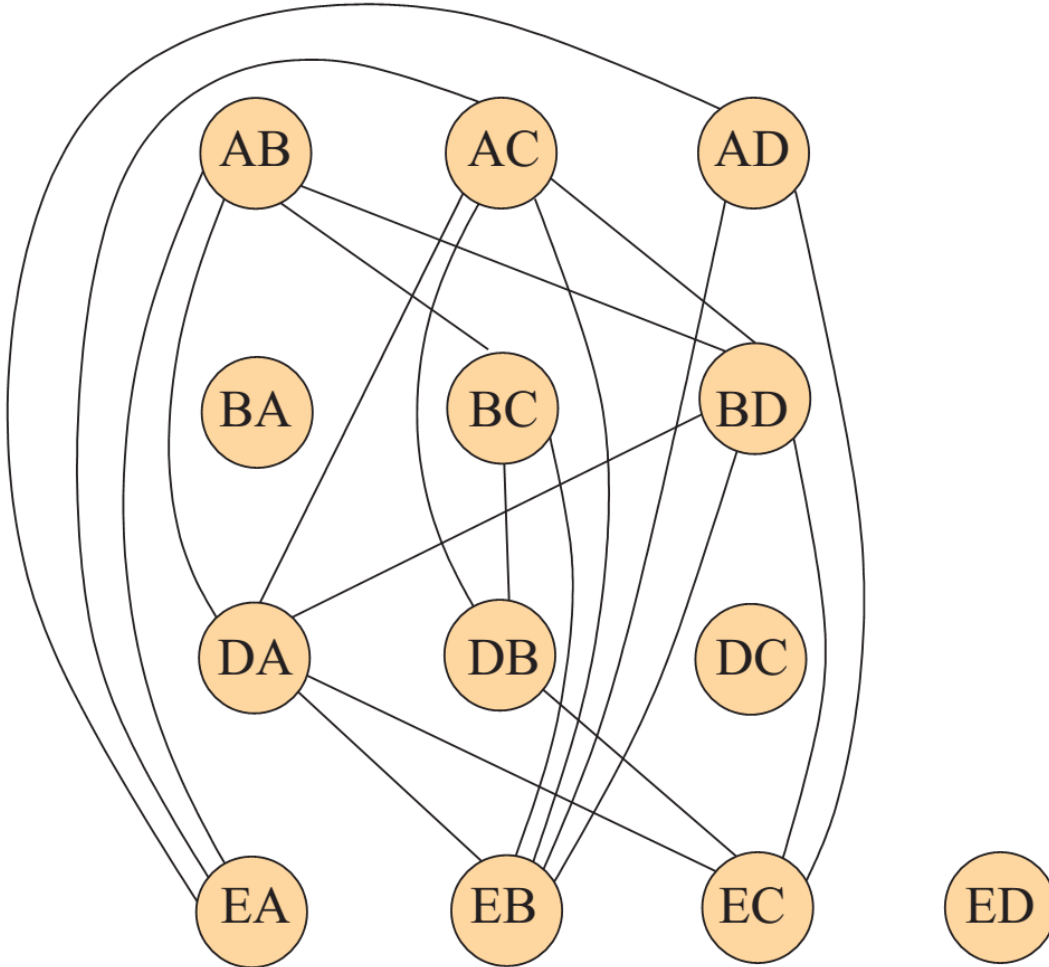
مثال

○ مدل انتزاعی مسئله: گردش‌ها و تلاقی بین آنها را می‌توان با یک گراف نشان داد که آن را گراف تقاطع می‌نامیم.

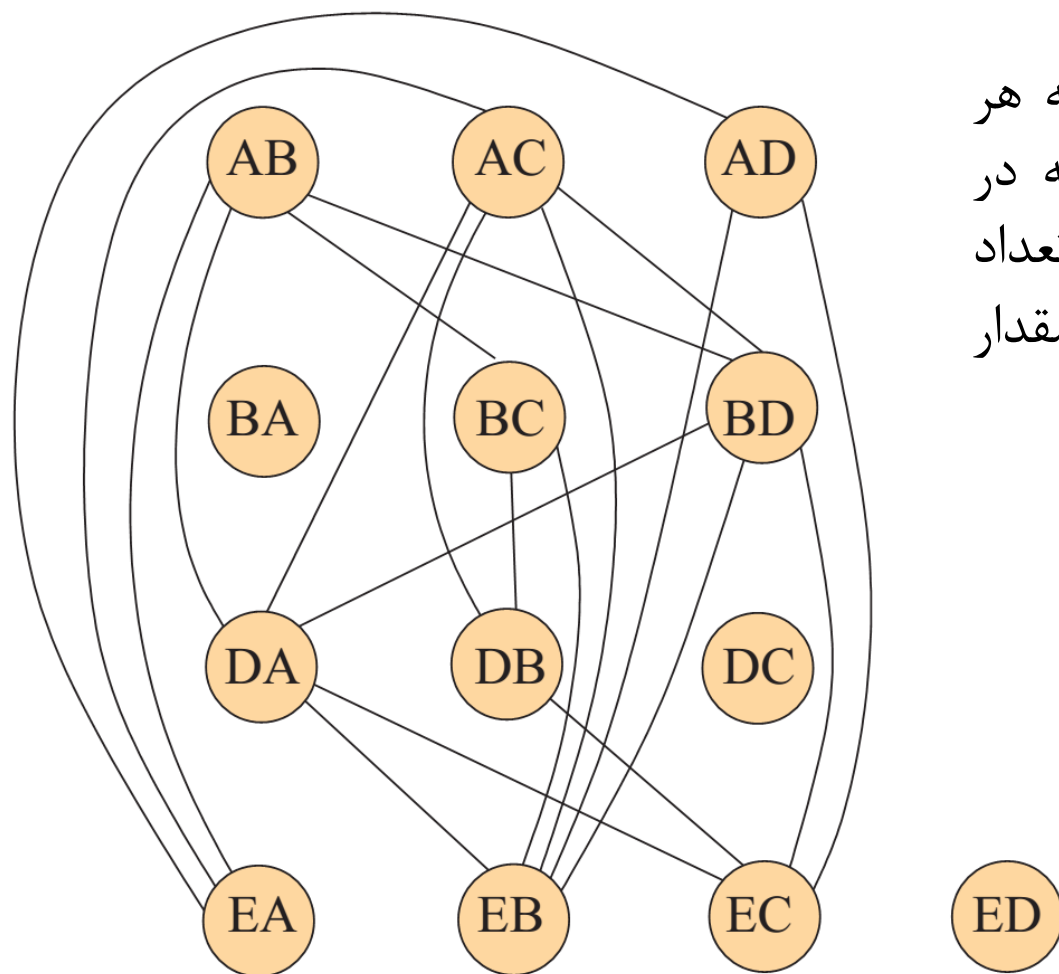


مثال

○ برای سادگی کار فرض می‌کنیم چراغ مورد نظر k زمان دارد و با رنگ‌های ۱ تا k مشخص شده است.



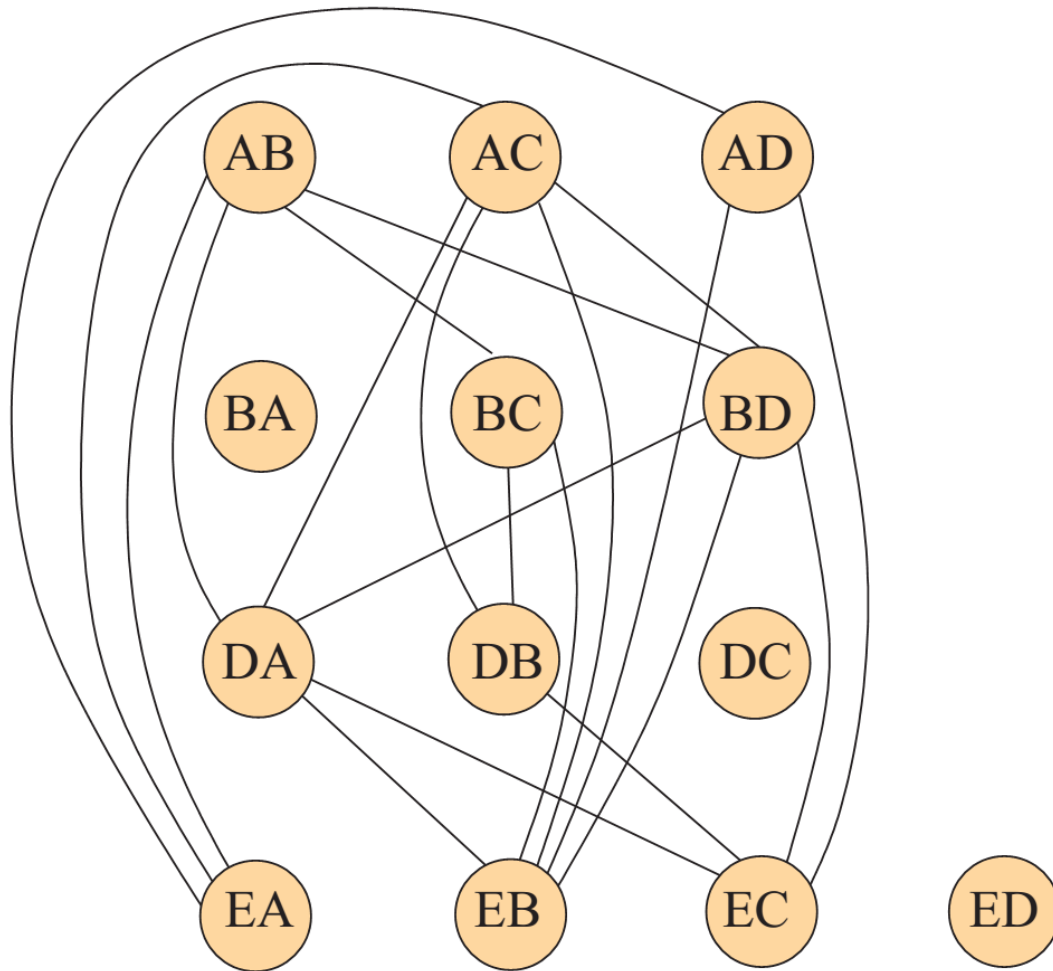
مثال



○ در مدل گراف، مسئله، انتصاب یک یا چند رنگ به هر راس گراف تقاطع است به طوری که دو راسی که در انتهای یک یال هستند رنگ یکسان نداشته باشند. تعداد این رنگ‌ها همان k است و ما می‌خواهیم کمترین مقدار آن را به دست بیاوریم

مثال

○ یافتن جواب بهینه (رنگ آمیزی گراف)



NP-Hard

مثال

○ یک الگوریتم حریصانه

CROSS-SOLUTION (G)

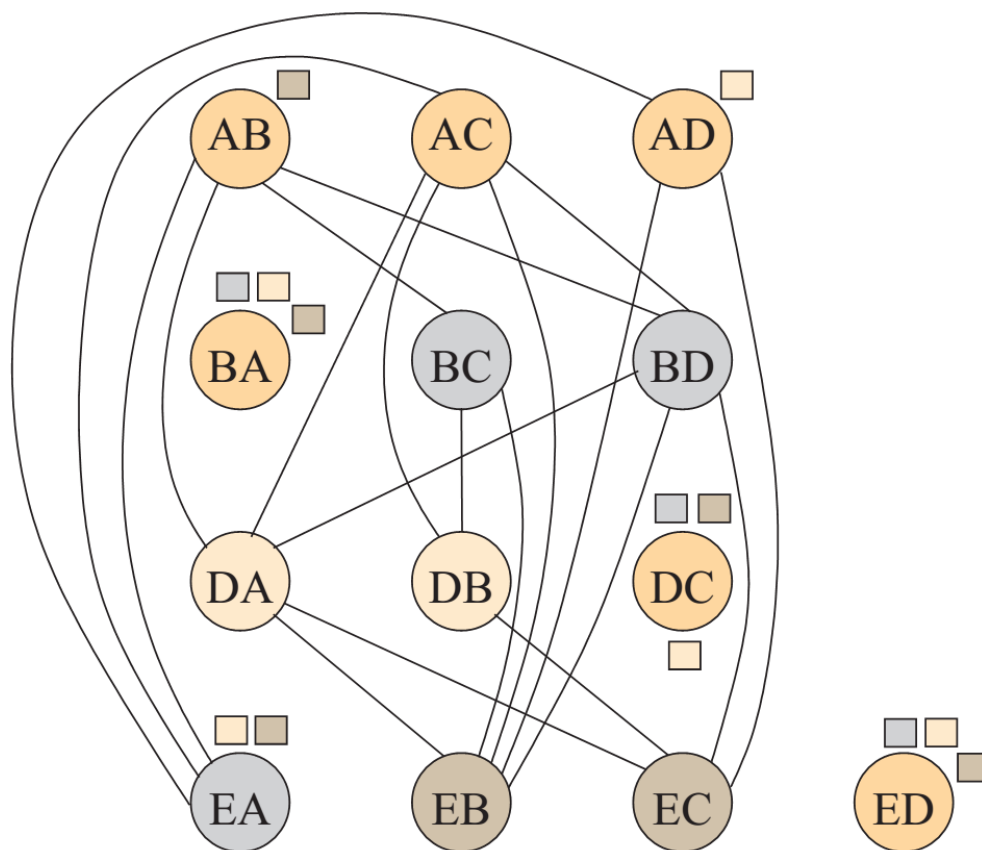
- ▷ Input: G گراف
 - ▷ Output: مجموعه‌ای از رأس‌ها برای هر رنگ؛ رأس‌هایی که آن رنگ را دارند
 - ▷ $VerticesWithColor$ و $ColorNo$: متغیرهای محلی: آرایه‌ی
- 1 MAKEGRAPH(G)
 - 2 $ColorNo \leftarrow 0$
 - 3 **while** رأس رنگ نشده در G وجود دارد
 - 4 **do** $ColorNo \leftarrow ColorNo + 1$
 - ▷ همه‌ی رأس‌هایی را که بتوان با $ColorNo$ رنگ کرد به دست آور
 - 5 $VerticesWithColor[ColorNo] \leftarrow$ GREEDYSOLUTION (G)
 - 6 **return** $VerticesWithColor$

GREEDYSOLUTION (G)

- 1 MAKEEMPTYSET ($newcolor$) ▷ مجموعه‌ی تهی $newcolor$ را ایجاد کن
- 2 **for** هر رأس v در G
- 3 **do if** v رنگ نشده است
- 4 **then if** v مجاور هیچ رأسی در $newcolor$ نباشد
- 5 **then** به v برچسب «رنگ‌شده» بزن
- 6 v را به مجموعه‌ی $newcolor$ اضافه کن
- 7 **return** $newcolor$

مثال

نتیجه انجام الگوریتم حریصانه



مثال

○ ساختمان/ساختار داده

- $\text{MAKEEMPTY}(G)$: ایجاد یک گراف تهی،
- $\text{INSERTVERTEX}(G, v)$ و $\text{INSERTEDGE}(G, u, v)$: درج یک رأس یا یک یال در G ،
- $\text{NUMEDGES}(G)$ ، $\text{NUMVERTICES}(G)$: تعداد رأس‌ها و یال‌های G را برمی‌گرداند،
- $\text{LISTOFVERTICES}(G)$: ترتیبی از رأس‌ها را تولید می‌کند،
- $\text{ADJACENTVERTICES}(G, v)$: رأس‌های مجاور یک رأس v را تولید می‌کند،
- $\text{COLORVERTEX}(G, v)$: رأس v را رنگ می‌کند،

داده گونه انتزاعی (ADT)

○ توصیف ریاضی/انتزاعی یک داده به همراه مجموعه عملگرهای آن

- ورودی و خروجی ها را تعریف می کند.
- از جزئیات و پیاده سازی حرفی نمیزند.

○ پیاده سازی:

- پیاده سازی شده یک ADT، ساختمان داده است.

○ کاربران، ساختمان داده را به صورت یک ADT می بینند.

مثال



○ ADT برای دستگاه نوشیدنی

- این دستگاه نوشیدنی ذخیره می کند.
- شامل عملگر درخواست یک نوشیدنی است.



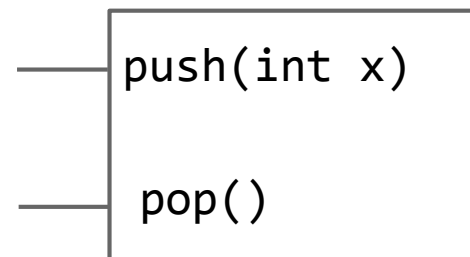
○ پیاده سازی:

- نوشیدنی چگونه ذخیره شده است.
- درخواست یک نوشیدنی چگونه انجام می شود.
- این موارد در داخل دستگاه است.

مثال

ADT پشته: ○

- عملگر push
- عملگر pop



7

مثال

○ ADT پشته:

- عملگر push
- عملگر pop

○ پیاده‌سازی:

- آرایه
- لیست پیوندی

مرور

○ ساختمان‌های داده:

- آرایه
- لیست
- صف
- پشته
- صف اولویت
- جدول درهم‌ساز
- درخت
- گراف

○ الگوریتم:

- مرتب‌سازی
- جستجو
- مربوط به گراف

○ تکنیک‌ها و مباحث نظری:

- تحلیل الگوریتم‌ها
- پیچیدگی
- نمادهای مجانبی
- حل بازگشتی