

# برنامه‌ی زمانبندی کلاس‌ها

استاد: دکتر مهرنوش شمس فرد  
صالح جعفری زاده - ۹۲۲ ۱۳۰۶۱



# تعریف پروژه

- هدف از اجرای این برنامه زمانبندی ترم آموزشی بر اساس داشته های زیر است:
  - جدول چینش دروس در ترم های درسی (چارت آموزشی)
  - تعداد واحد های لازم برای هر درس
  - درس های ارائه شده در این ترم
  - برنامه زمانی اساتید برای حضور در دانشکده
  - درس هایی که هر یک از اساتید تمایل به تدریس آن دارند
- شرط هایی که باید برای این زمانبندی در نظر گرفته شوند:
  - درس هایی که در یک ترم ارائه می شوند تلاقی نداشته باشند
  - در هر ساعت هر استاد می تواند فقط در یک کلاس تدریس کند

# شرح راه حل



- برای حل این مساله پارامترهای زیر را به عنوان متغییر میگیریم:
  - اساتید
    - درس متمایل به ارائه ی آن
    - زمان های حضور در دانشکده
  - درس
    - ترم ارائه ی آن
    - تعداد واحد
  - بازه های زمانی
    - روز و ساعت شروع و پایان
    - ارزش آن بازه



## شرح راه حل

- الگوریتم بدین گونه است که در هر مرحله به یک درس استاد و بازه‌ی زمانی منطبق می‌کنیم و به سراغ درس‌های بعدی می‌رویم تا تمامی دروس مقدار گیرند.
- اگر در مرحله‌ای هیچ مقدار مجازی برای تطابق با یک درس پیدا نکردیم به مرحله قبل بازگشته و با انتخاب مقداری دیگر برای آن مرحله الگوریتم را ادامه می‌دهیم.
- در هر مرحله الگوریتم جایگشت‌های ممکن اساتید و بازه‌های زمانی را برای یک درس در نظر می‌گیرد و با استفاده از الگوریتم‌هایی مکاشفه‌ای که بعدتر به توضیح آن می‌پردازیم یک مقدار را انتخاب می‌کند و به درس منطبق می‌کند.

# الگوریتم های مکشافه ای

## • پیاده سازی

- برای پیاده سازی این الگوریتمها از PriorityQueue استفاده شده است.
- مقادیر مجاز برای تطابق با یک درس در صف validPermutationQ ریخته میشود.
- دروسی که در صف نوبت دهی هستند در courseAssigningQ ریخته میشوند.
- هرکدام با استفاده از Comparator ای که در کلاس Heuristics دارند اولویت دهی و انتخاب میشوند

# الگوریتم های مکشافه‌ای

## • الگوریتم مکشافه‌ای انتخاب درس بعدی

- در هر مرحله درسی انتخاب می‌شود که تعداد استاددهای کمتری آن درس ارائه می‌کنند.

دلیل: هر چه که انتخاب یک درس به تعویق بیافتد به معنی آن است که این درس در لایه‌های نزدیکتر به برگ مقداردهی میشود و هرچه که مقدار مجاز برای آن بیشتر باشد به ازای یکبارتغییر آن همه ی فرزندانیش نیز باید بررسی شوند پس هرچه در لایه‌های پایین تر باشد تعداد بررسی کمتری را ایجاب می‌کند.

```
PriorityQueue<Course> courseAssigningQ=new PriorityQueue(Heuristics.teacherAssigningToCourseComparator);
```

```
public class Heuristics {

    public static Comparator<Course> teacherAssigningToCourseComparator=new Comparator<Course>() {
        @Override
        public int compare(Course o1, Course o2) {
            try {
                return o1.teacherOfThisCourse.size()- o2.teacherOfThisCourse.size();
            }catch (Exception e){
                System.err.println(o1+" "+o2);
            }
            return 0;
        }
    }
}
```

## • تاثیر مکشافه

- زمان پاسخگویی با مکشافه نزدیک به ۱ دقیقه
- زمان پاسخگویی بدون مکشافه بیش از ۱۰ دقیقه



# الگوریتم های مکشافه‌ای

- الگوریتم مکشافه‌ای انتخاب استاد و زمان برای یک درس
- در این مرحله بین تمامی جایگشت های موجود از اساتید و زمان های ارائه هر درس جایگشتی انتخاب میشود که زمان آن در بازه ای باشد که دروس کمتری تا اینجا به آن منطبق شده اند.  
دلیل: در اینصورت امکان تلاقی در مراحل بعد کمتر میشود.
- البته برای محاسبه مقدار نهایی میانگین مقدار بالا با ارزش هر بازه در نظر گرفته می‌شود که در اسلاید بعد راج به ارزش دهی بازه ها صحبت خواهد شد.

```
PriorityQueue<AssignedCourse> ValidPermutationQ=new PriorityQueue(new Comparator() {
    @Override
    public int compare(Object o1, Object o2) {
        int sum1=0;
        int sum2=0;
        for(AssignedCourse w:assignedCourses)
            if(w.timeParts.contains(((AssignedCourse)o1).timeParts.get(0)))sum1++;
        for(AssignedCourse w:assignedCourses)
            if(w.timeParts.contains(((AssignedCourse)o2).timeParts.get(0)))sum2++;
        int a=sum2-sum1;
        int b= ((AssignedCourse)o1).timeParts.get(0).id-((AssignedCourse)o2).timeParts.get(0).id;
        return (a+b)/2;
    }
});
```

- تاثیر مکشافه
- زمان پاسخگویی با مکشافه نزدیک به ۳ دقیقه
- زمان پاسخگویی بدون مکشافه بیش از ۱۰ دقیقه



# ارزش گذاری بازه‌ها (بخش امتیازی)

- بازه‌های زمانی طبق جدول زیر ارزش گذاری میشوند:
- مزیت اول: درس‌ها در زمان‌های سبز رنگ مقدار دهی میشوند.
- مزیت دوم: درس‌های یک ترم نزدیک به هم می‌افتند و زمان خالی دانشجو برای فعالیت‌های دیگر بیشتر میشود.

روز/ساعت	۹-۷:۳۰	۱۰:۳۰-۹	۱۲-۱۰:۳۰	۱۴:۳۰-۱۳	۱۶-۱۴:۳۰	۱۷:۳۰-۱۶
شنبه	۲۶	۱	۲	۳	۴	۲۱
یکشنبه	۲۷	۵	۶	۷	۸	۲۲
دوشنبه	۲۸	۹	۱۰	۱۱	۱۲	۲۳
سه‌شنبه	۲۹	۱۳	۱۴	۱۵	۱۶	۲۴
چهارشنبه	۳۰	۱۷	۱۸	۱۹	۲۰	۲۵

زمان اجرای کد با استفاده از هر دو الگوریتم مکاشفه ای برای زمانبندی ۵ ترم آموزشی زیر ۵ ثانیه است!



# خروجی برنامه



- برنامه بعد از زمانبندی کلاس ها با استفاده از کتابخانهی `org.apache.poi` خروجی را به تفکیک ترم در فایل اکسل می‌ریزد.

	A	B	C	D	E	F	G
1	Days	7:30 - 9	9 - 10:30	10:30 - 12	13:00 - 14:30	14:30 - 16	16 - 17:30
2	SAT	DS:Mohsen Ebrahimimoghadam	DL1:Binesh Marvasti ,	DL1:Binesh Marvasti ,		AutomataB:Ramak Ghavamizade ,	
3	SUN		DS:Mohsen Ebrahimimoghadam	ASM:Ahmad Mahmoodi ,	L1:Asatid GA1 ,		
4	MON			Automata:Ramak Ghavamizade ,			
5	TUE			ASM:Ahmad Mahmoodi ,			
6	WED		AutomataB:Ramak Ghavamizade	Automata:Ramak Ghavamizade ,			
7							

## توسعه و انعطاف پذیری برنامه

- برنامه به گونه ای طراحی شده که به ازای درس‌هایی که بیشتر از ۲ بازه زمانی در هفته احتیاج دارند و به ازای تعداد ترم‌های بیشتر قادر به برنامه‌ریزی است.
- با اندک تغییر کد می‌توان امکان Back Jump را نیز به الگوریتم اضافه کرد.
- به علت کمبود وقت این قسمت پیاده‌سازی نشده.
- مشکل افتادن بازه‌های زمانی مربوط به درس پشت سرهم در صورت داشتن پیش‌فرض‌هایی مثل اینکه حداقل فاصله ارائه آن دو بازه چقدر باشد و زوجیت روز ارائه حفظ شود یا نه با چند دستور شرطی قابل حل شدن است.

“ چنید گُفت پیغمبر راست گوی  
ز لَهواره تا گور دانش بجوی ”

حکیم ابولقاسم فردوسی

با تشکر از همراهیتان!

