

طراحی الگوریتم ها

جلسه اول

طرح درس

• اهداف:

- آشنایی با روش های تحلیل الگوریتم
- آشنایی با نحوه اثبات درستی الگوریتم
- آشنایی با روش ها متداول طراحی الگوریتم
- آشنایی با الگوریتم های متداول در زمینه مهندسی و علوم کامپیوتر

کتابہا

- Textbook:
 - **Cormen**, Thomas H., Charles E. **Leiserson**, Ronald L. **Rivest**, and Clifford **Stein**. Introduction to Algorithms. 3rd. ed. Cambridge, MA: MIT Press.
- References:
 - **Horowitz**, E., **Sahni**, S Fundamentals of Computer Algorithm. Computer science press. 2009
 - Richard **Neapolitan**, and Kumarss **Naimipour**. Foundations of algorithms. Jones & Bartlett Learning, 2010.

ارتباط و تدریس یاری T.A.

- روزهای چهارشنبه ساعت ۳ تا ۴.۵
- اطلاع رسانی در شبکه اجتماعی t.me/DAAAUT
- اطلاع رسانی در شبکه اجتماعی courses.aut.ac.ir
- برگزاری کلاس مجازی lms.aut.ac.ir
- ارتباط با من:

— ایمیل hoorfarhamid@yahoo.com

— شبکه اجتماعی [@hamidhoorfar](https://www.instagram.com/hamidhoorfar)

عنوان مباحث

1. مقدمه و نقش الگوریتم ها (فصل ۱ و ۲)
2. مرور رشد توابع و مفهوم سائز مسئله (فصل ۳)
3. حل رابطه بازگشتی، درخت بازگشت و قضیه اصلی (فصل ۴)
4. مرتب سازی پشته ای، مرتب سازی سریع، مرتب سازی درجی، اثبات درستی مرتب سازی درجی و تحلیل حالت متوسط زمان (فصل ۶ و ۷)
5. مرتب سازی با زمان خطی (فصل ۸)
6. پیدا کردن میانه و تحلیل سرشکن (فصل ۹ و ۱۷)
7. روش تقسیم و حل – موزائیک، استراسن، تورنمنت بازی (فصل ۴)
8. برنامه ریزی پویا (فصل ۱۵) (۲ هفته)
9. روش حریصانه (فصل ۱۶) (۲ هفته)

عنوان مباحث

10. روش عقبگرد و روش شاخه و حد
11. مروری سریع بر الگوریتم های گراف و درخت
12. الگوریتم جریان بیشینه (فصل ۲۶)
13. نظریه پیچیدگی محاسبات (فصل ۳۴)

بارم بندی نمره نهایی (غیر قطعی)

• ۷ نمره پایان ترم

• ۵ نمره میان ترم

• ۸ نمره تمرینات

• ۱ نمره پروژه

• ۹ نمره پایان ترم

• ۶ نمره میان ترم

• ۵ نمره تمرینات

• ۲ نمره پروژه

نقش الگوریتم ها در محاسبه

- تعریف الگوریتم:

- یک سلسله مراتب از گام های محاسباتی خوش تعریف.
- مقدار یا مجموعه مقادیر را بعنوان Input دریافت می کند
- مقدار یا مجموعه مقادیر را بعنوان output تولید می کند

مثال متداول از الگوریتم ها برای مسائل محاسباتی

- Sorting problem
- Searching problem

Sorting problem

sorting problem:

Input: A sequence of n numbers $\langle a_1, a_2, \dots, a_n \rangle$.

Output: A permutation (reordering) $\langle a'_1, a'_2, \dots, a'_n \rangle$ of the input sequence such that $a'_1 \leq a'_2 \leq \dots \leq a'_n$.

Instance of sorting problem:

Input: $\langle 31, 41, 59, 26, 41, 58 \rangle$

Output: $\langle 26, 31, 41, 41, 58, 59 \rangle$

- الگوریتم صحیح برای هر نمونه از ورودی، با خروجی صحیحی متوقف میشود.
به چنین الگوریتمی گفته میشود که مساله محاسباتی را حل کرده است.

مسایل قابل حل توسط الگوریتم ها

- الگوریتم های تحت اینترنت
- الگوریتم های مرتبط با تجارت الکترونیکی:
 - توانایی نگهداری اطلاعات: اعداد کارت اعتباری، کلمات عبور و رمزها، صورت حسابها، رمزگشایی،
 - امضاهای دیجیتال و ...
- الگوریتمهای مرتبط با صنعت:
 - مکان یابی چاههای نفت
 - خطوط هوایی
 - پیش بینی انتخابات
- الگوریتم های مسیریابی:
 - یافتن کوتاهترین مسیر
- الگوریتم های محاسباتی یافتن توالی بهینه ضرب ماتریس ها
- الگوریتم یافتن *convex hull*

efficient algorithms and Hard Problem

- یکی از اهداف مهم این درس طراحی الگوریتم های کارآمد است.
 - مقیاس کارایی سرعت است : تا چه اندازه یک الگوریتم زمان صرف می کند تا نتایج خود را تولید کند.
 - *Hard problem*
 - مسائل محاسباتی از لحاظ مقیاس کارایی به دسته های مختلف تقسیم می شوند.
 - دسته ای از مسائل که تاکنون ثابت نشده است که برای آنها الگوریتم کارآمد وجود دارد، *NP-complete* گفته می شوند.
 - اگر الگوریتم کارآمدی برای یکی از آنها موجود باشد آنگاه الگوریتم کارآمد برای مابقی آنها نیز حتما موجود است.
 - این مسائل بسیار شبیه مسائلی هستند که برای آنها الگوریتم کارآمد وجود دارد.
- کارآمد ؟؟؟؟؟؟؟

یک نمونه مساله NP-complete

- *Travelling salesman problem (TSP)*

- ابتدا در سده ۱۸ مسائل مربوط به آن توسط ویلیام همیلتون و چوریو مطرح شد و سپس در دهه ۱۹۳۰ شکل عمومی آن به وسیله ریاضیدانانی مثل کارل منگر از دانشگاه هاروارد و هاسلر ویتنی از دانشگاه پرینستون مورد مطالعه قرار گرفت.
- شرح مسئله بدین شکل است که:

— تعدادی شهر داریم و هزینه رفتن مستقیم از یکی به دیگری را می دانیم. مطلوب است کم هزینه ترین مسیری که از یک شهر شروع شود و از تمامی شهرها دقیقاً یکبار عبور کند و به شهر شروع بازگردد.

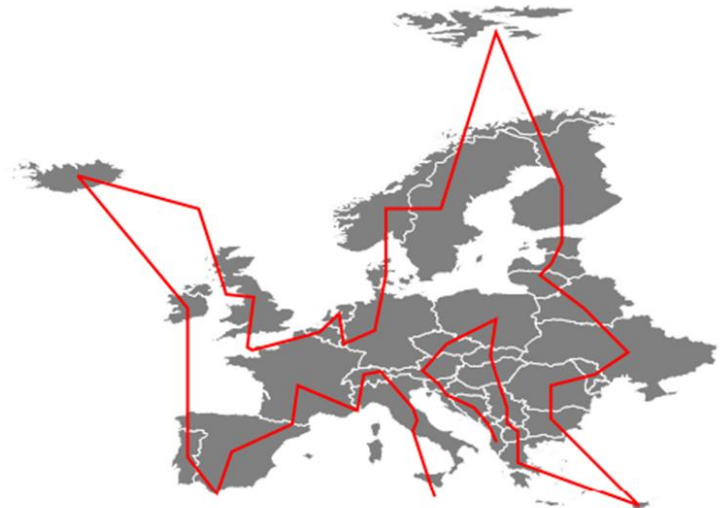
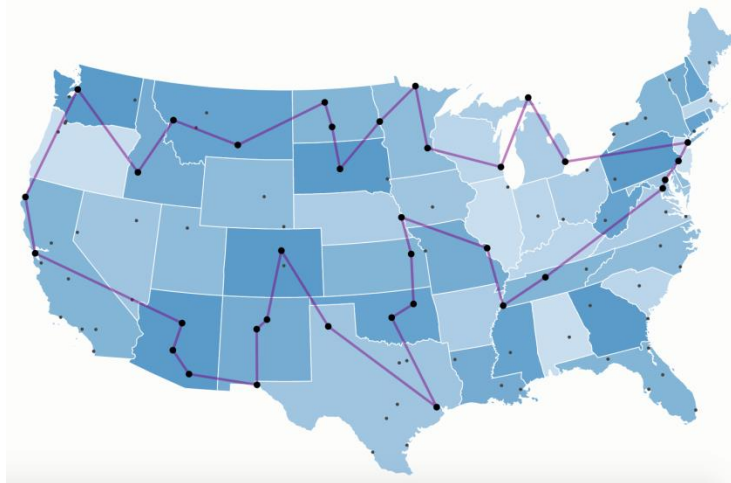


Figure 1 Geographical Routes as per TSP

تکنولوژی الگوریتم ها

- افزایش سرعت : زمان محاسبه یک منبع محدود است.
- کاهش مصرف حافظه : حافظه ارزان است ولی رایگان نیست.
- تولید الگوریتم های پایان پذیر

Time efficiency

- تفاوت در راندمان الگوریتم ها علاوه بر وابستگی به سخت افزار و نرم افزار به نحوه طراحی الگوریتم ها نیز وابسته است.
- مقایسه نحوه طراحی الگوریتم ها و کارایی آنها
- $Insertion\ sort \rightarrow c_1 n^2 \rightarrow 2n^2$
- $Merge\ Sort \rightarrow c_2 n \lg n \rightarrow 50n \lg n$
- مرتب سازی ده میلیون عدد:
- $Computer\ A \rightarrow 10000000000\ clock/sec = 10^{10}$
- $Computer\ B \rightarrow 10000000\ clock/sec = 10^7$

$$\frac{2 \cdot (10^7)^2 \text{ instructions}}{10^{10} \text{ instructions/second}} = 20,000 \text{ seconds (more than 5.5 hours) ,}$$

$$\frac{50 \cdot 10^7 \lg 10^7 \text{ instructions}}{10^7 \text{ instructions/second}} \approx 1163 \text{ seconds (less than 20 minutes)}$$

تمرین

- کوچکترین مقدار n که به ازای آن الگوریتمی با زمان اجرای $100n^2$ سریعتر از الگوریتمی با زمان اجرای 2^n روی ماشین یکسان اجرا میشود چیست؟

n	$100 n^2$	2^n	
1	100	2	>
2	400	4	>
3	900	8	>
4	1600	16	>
5	2500	32	>
6	3600	64	>
7	4900	128	>
8	6400	256	>
9	8100	512	>
10	10000	1024	>
11	12100	2048	>
12	14400	4096	>
13	16900	8192	>
14	19600	16384	>
15	22500	32768	<
16	25600	65536	<
17	28900	131072	<
18	32400	262144	<
19	36100	524288	<
20	40000	1048576	<

پاسخ