HW-04

Algorithm Design - Spring 2026

Instructor: Dr. Amin Eskandari

Head Teaching Assistants: Hamid Namjoo, Amir Hossein Hemati

Teaching Assistants: Ali Nikvan, Viyana Hasanbeigi, Nima Soltani, Reza Rezaie, Maryam Meidanshahi , Babak Janfaza, Elham Mosavi

---

سوال ۱ :

📡🚀نبرد با نویزهای فضایی و چالش توان سریع

یک کاوشگر فضایی در حال ارسال سیگنال به زمین است.

قدرت اولیه‌ی سیگنال برابر با عدد صحیح **a**است.

این سیگنال برای عبور از لایه‌های مختلف نویز تقویت می‌شود و پس از عبور از **nلایه‌ی نویز**، قدرت نهایی آن برابر خواهد بود با:

$$a^n$$

اما یک مشکل وجود دارد...

به دلیل محدودیت‌های سخت‌افزاری و زمانی، امکان محاسبه‌ی توان با ضرب تکراری عدد **a** به‌صورت n بار وجود ندارد.

عدد **n** می‌تواند بسیار بزرگ باشد (تا 10^9)و حتی ممکن است برابر با صفر باشد.

بنابراین باید از یک روش کارآمد استفاده شود.

🎯مأموریت شما

تابعی بنویسید که با استفاده از روش **تقسیم و حل(Divide and Conquer)** ، مقدار a^nرا در زمان **O(log n)** محاسبه کند.

📥ورودی

ورودی شامل دو خط است:

a

n

- خط اول: یک عدد صحیح **a** (پایه‌ی توان)
- خط دوم: یک عدد صحیح نامنفی **n** (توان)
- تضمین می‌شود:

$$0 \leq x \leq 10^9$$

📤 خروجی

یک عدد صحیح که نشان‌دهنده‌ی مقدار:

a^n

است.

📌 قوانین مسئله

- مقدار **n = 0** باید به‌درستی مدیریت شود.
- استفاده از ضرب تکراری ساده (n بار ضرب کردن) مجاز نیست.
- پیچیدگی زمانی الگوریتم باید **O(log n)** باشد.
- تمام مقادیر ورودی صحیح هستند.

نمونه ورودی و خروجی

ورودی:

2
10

خروجی:

1024

سوال ۲ :

🛰️📡رادار شناسایی جنگنده‌های رادارگریز

نیروی هوایی متوجه ورود تعدادی پهپاد ناشناس به حریم کشور شده است.
موقعیت هر پهپاد به‌صورت یک نقطه در صفحه‌ی دوبعدی ثبت شده است.
برای تحلیل رفتار این پهپادها، فرماندهی نیاز دارد **کمترین فاصله بین هر دو پهپاد** را محاسبه کند.

اما یک چالش مهم وجود دارد...

تعداد پهپادها می‌تواند بسیار زیاد باشد (تا 10^5) و مختصات آن‌ها نیز ممکن است مقادیر بزرگی داشته باشند (تا 10^9) .
بنابراین استفاده از روش‌های ساده با پیچیدگی بالا (مانند بررسی همه‌ی جفت‌ها با روش brute-force) قابل قبول نیست.

🎯مأموریت شما

با استفاده از روش **تقسیم و حل (Divide and Conquer)** برنامه‌ای بنویسید که کمترین فاصله بین دو نقطه را در زمان:

$$O(n \log n)$$

محاسبه کند.

⚠️راه‌حل‌های brute-force نمره کامل دریافت نمی‌کنند.
وجود حداقل دو نقطه تضمین شده است.

📥ورودی

n

x1 y1

x2 y2

...

xn yn

- خط اول: عدد صحیح **n** (تعداد پهپادها)

$$2 \leq n \leq 10^5$$

- در هر یک از n خط بعدی: دو عدد صحیح **x** و **y** که مختصات یک پهپاد را نشان می‌دهند.

$$-10^9 \leq x, y \leq 10^9$$

📤خروجی

یک عدد اعشاری که کوچک‌ترین فاصله بین دو پهپاد را نشان می‌دهد.

- خروجی باید با **دقیقاً دو رقم اعشار** چاپ شود.

📌قوانین مسئله

- استفاده از الگوریتم Divide and Conquer الزامی است.
- پیچیدگی زمانی باید $O(\mathrm{n}\log n)$باشد.
- وجود حداقل دو نقطه تضمین شده است.
- پاسخ باید فاصله اقلیدسی بین نزدیک‌ترین دو نقطه باشد.
- فرمت خروجی باید دقیقاً با دو رقم اعشار باشد.

نمونه ورودی و خروجی

ورودی:

2
0 0
3 4

خروجی:

5.00

سوال ۳ :

✨💻ابرکامپیوتر پردازش تصویر – چالش ضرب ماتریس استراسن

شرکت "**پیکسل‌تک** "در حال طراحی یک پردازنده جدید برای گرافیک بازی‌های سنگین است.
در قلب این پردازنده، باید **دو ماتریس بزرگ که حاوی اطلاعات نوری پیکسل‌ها هستند** در هم ضرب شوند.

مهندسان متوجه شده‌اند که ضرب معمولی سطر در ستون باعث داغ شدن بیش از حد چیپ می‌شود.
بنابراین نیاز به یک الگوریتم هوشمند برای کاهش تعداد ضرب‌ها دارند.

🎯مأموریت شما

با استفاده از **الگوریتم استراسن(Strassen Algorithm)** ، حاصل‌ضرب دو ماتریس مربعی را محاسبه کنید به گونه‌ای که:

- تعداد ضرب‌های بازگشتی از ۸ به ۷ کاهش یابد
- الگوریتم **بازگشتی Divide and Conquer** باشد
- ساختار بازگشتی الگوریتم به درستی رعایت شود

💡نکته:

- برایn های کوچک، می‌توانید از ضرب معمولی (حالت پایه) استفاده کنید
- **تابع آماده ضرب ماتریس) مثل (numpy.dot مجاز نیست**

📥ورودی

n
A11 A12 ... A1n
...
An1 An2 ... Ann
B11 B12 ... B1n
...
Bn1 Bn2 ... Bnn

- خط اول: عدد صحیح **n**که بعد ماتریس‌هاست$(n = 2^k)$
- nخط بعد: عناصر ماتریس اول
- nخط بعد: عناصر ماتریس دوم

📤خروجی

C11 C12 ... C1n

...

Cn1 Cn2 ... Cnn

- nخط، که هر خط شامل n عدد صحیح است
- نشان‌دهنده ماتریس حاصل‌ضرب دو ماتریس ورودی با **الگوریتم استراسن**

📌قوانین مسئله

- nهمیشه توان دو است$(n = 2^k)$
- برایn های کوچک می‌توان ضرب معمولی را به عنوان حالت پایه استفاده کرد
- استفاده از **تابع آماده ضرب ماتریس مجاز نیست**
- الگوریتم باید بازگشتی و مبتنی بر **Divide and Conquer**باشد
- رعایت کاهش ضرب‌های بازگشتی از ۸ به ۷ الزامی است

نمونه ورودی و خروجی

ورودی:

1
5
7

خروجی:

35

سوال ۴ :

📜🏺رمزگشایی از کتیبه باستانی

باستان‌شناسان در **تخت‌جمشید** کتیبه‌ای طولانی پیدا کرده‌اند که شامل هزاران کاراکتر است.
آن‌ها حدس می‌زنند که یک" **الگوی مقدس** "(یک عبارت خاص) چندین بار در این کتیبه تکرار شده است.
اگر بتوانید **تمام محل‌های دقیق شروع این الگو** را در متن پیدا کنید، رمز مخفی کتیبه آشکار خواهد شد.
اما یک چالش وجود دارد:
متن کتیبه بسیار طولانی است و روش‌های ساده جستجو، حتی با کامپیوترهای امروزی، زمان زیادی می‌برند.

🎯مأموریت شما

با استفاده از الگوریتم **KMP (Knuth-Morris-Pratt)**، تمام اندیس‌هایی که الگو در آن‌ها شروع می‌شود را پیدا کنید.

💡نکات مهم:

- **استفاده از find، یا regex مجاز نیست**
- الگوریتم باید دارای پیچیدگی زمانی **O(n + m)**باشد، که n طول متن و m طول الگو است

📥ورودی

text

pattern

- خط اول: متن کتیبه (رشته‌ای از کاراکترها)
- خط دوم: الگوی مورد نظر

📤خروجی

- اندیس‌های شروع تمام رخدادهای الگو در متن، به صورت **0-based**
- اندیس‌ها باید با **یک فاصله** از هم چاپ شوند
- در صورت نبود هیچ رخداد از الگو، چاپ شود:

Null

📌قوانین مسئله

- الگوریتم **KMP** الزامی است
- پیچیدگی زمانی **O(n + m)** باید رعایت شود
- اندیس‌ها **0-based** هستند
- چاپ فاصله بین اندیس‌ها اجباری است
- عدم استفاده از توابع آماده جستجو(find, in, regex)

نمونه ورودی و خروجی

ورودی:

aaaaa
aa

خروجی:

0 1 2 3