

سوال ۱ :

🏠🕒 چالش زمان‌بندی سفارش‌ها در کارگاه تولیدی

یک کارگاه تولیدی کوچک، سفارشی برای ساخت ۵ قطعه خاص دریافت کرده است.

فرآیند تولید هر قطعه به این صورت است که:

- ابتدا باید روی دستگاه برش (Machine 1) پردازش شود
- سپس روی دستگاه پرداخت (Machine 2) تکمیل گردد

به دلیل محدودیت‌های فضا و هزینه، در هر لحظه هر دستگاه فقط می‌تواند یک قطعه را پردازش کند.

همچنین ترتیب انجام قطعات روی هر دو دستگاه باید یکسان باشد.

مدیر کارگاه از شما—به‌عنوان دانشجوی درس طراحی الگوریتم—درخواست کرده است که با استفاده از

الگوریتم جانسون، ترتیب اجرای قطعات را به‌گونه‌ای تعیین کنید که زمان کل تکمیل سفارش‌ها

(Makespan) حداقل شود.

📌 اطلاعات ورودی

زمان پردازش هر قطعه (بر حسب دقیقه) در جدول زیر داده شده است:

قطعه	Machine 1	Machine 2
A	6	2
B	4	7
C	9	3
D	5	8
E	7	4

الف) الگوریتم جانسون را به صورت **مرحله به مرحله** روی داده‌های فوق اجرا کنید و ترتیب نهایی انجام قطعات را به دست آورید.
(مراحل انتخاب، مقایسه و جایگذاری قطعات را به طور کامل توضیح دهید.)

ب) با استفاده از ترتیب به دست آمده در قسمت (الف):

- نمودار زمانی اجرای قطعات روی هر دو ماشین را رسم کنید
 - **زمان کل تکمیل سفارش‌ها (Makespan)** را محاسبه نمایید
- ♦ رسم نمودار به صورت **جدول زمانی** نیز قابل قبول است.
-

ج) یک **شبه‌کد** بنویسید که:

- ورودی آن شامل لیستی از کارها و زمان پردازش آن‌ها روی دو ماشین باشد
 - خروجی آن، **ترتیب اجرای بهینه کارها طبق الگوریتم جانسون** را تولید کند
-

د) مدیر کارگاه سؤال می‌کند:

اگر یک **دستگاه سوم** به خط تولید اضافه شود، آیا همچنان می‌توان از الگوریتم جانسون استفاده کرد؟ چرا؟
♦ پاسخ را **کوتاه، مفهومی و دقیق** بنویسید.

نمونه ورودی و خروجی

ورودی:

5

-3

خروجی:

سوال ۲ :

 چالش بهینه‌سازی مسیرها در شهر هوشمند

شهرداری یک شهر هوشمند تصمیم دارد سامانه‌ی مدیریت ترافیک را ارتقا دهد. این شهر از چندین تقاطع تشکیل شده که بین آن‌ها خیابان‌هایی با زمان‌های متفاوت عبور وجود دارد. به دلیل اجرای پروژه‌های عمرانی:

- همه‌ی خیابان‌ها دوطرفه نیستند
 - برخی مسیرها فقط در یک جهت خاص قابل استفاده‌اند
 - هدف شهرداری طراحی سیستمی است که بتواند:
 - کوتاه‌ترین زمان سفر بین هر دو تقاطع را محاسبه کند
 - و در صورت نبود مسیر قابل دسترس، این وضعیت را تشخیص دهد
- برای پیاده‌سازی این سیستم، از شما خواسته شده است از الگوریتم فلوید-وارشال استفاده کنید.

 اطلاعات ورودی

فرض کنید شهر شامل ۵ تقاطع با برچسب‌های زیر است:

A, B, C, D, E

زمان عبور مستقیم بین تقاطع‌ها (بر حسب واحد زمان) مطابق جدول زیر داده شده است (نماد ∞ به معنی نبود مسیر مستقیم است):

	A	B	C	D	E
A	0	3	∞	7	∞
B	∞	0	2	∞	∞
C	∞	∞	0	1	4

A B C D E

D ∞ ∞ ∞ 0 2

E ∞ ∞ ∞ ∞ 0

🎯 وظایف شما

(الف)

- جدول فوق را به صورت **ماتریس ورودی الگوریتم فلوید-وارشال** نمایش دهید.
- توضیح دهید مقدار ∞ در این الگوریتم چه مفهومی دارد و در پیاده‌سازی عملی چگونه باید با آن برخورد شود.

(ب)

- پس از در نظر گرفتن تقاطع C به عنوان رأس واسط:
- مقادیر به‌روزرسانی‌شده‌ی سطر مربوط به تقاطع A را محاسبه کنید.
 - مشخص نمایید آیا مسیر کوتاه‌تری از A به E به دست می‌آید یا خیر و دلیل آن را توضیح دهید.
- ♦ نیازی به اجرای کامل الگوریتم نیست.

(ج)

تابعی به زبان پایتون بنویسید که:

- یک **ماتریس وزن** را به عنوان ورودی دریافت کند
- الگوریتم **فلوید-وارشال** را اجرا نماید
- **ماتریس کوتاه‌ترین فاصله** بین تمام جفت رأس‌ها را خروجی دهد

فرض کنید:

- تعداد تقاطع‌ها حداکثر ۱۰۰ است
- مسیرهای غیرقابل دسترس با مقدار -1 مشخص می‌شوند

(د)

- پیچیدگی زمانی الگوریتم فلوید-وارشال را بنویسید و توضیح دهید چرا برای این مسئله مناسب است.
- اگر تعداد تقاطع‌ها به ۱۰,۰۰۰ افزایش یابد، آیا همچنان استفاده از این الگوریتم منطقی خواهد بود؟
پاسخ خود را با دلیل بیان کنید.

نمونه ورودی و خروجی

ورودی:

5 2
ABCDEF
AB 10
AC 20
BC 15
DEF 25
XYZ 30

خروجی:

55

سوال ۳ :

طراحی شبکه‌ی فیبر نوری بین ایستگاه‌های خدمات شهری

شهرداری یک شهر هوشمند قصد دارد میان n ایستگاه خدمات شهری (مانند آتش‌نشانی، اورژانس، پلیس و ...) یک شبکه‌ی ارتباطی فیبر نوری ایجاد کند.

بین برخی از ایستگاه‌ها امکان برقراری اتصال فیبر نوری وجود دارد و هزینه‌ی هر اتصال متفاوت است. با این حال، به دلیل محدودیت تجهیزات و منابع، برخی از اتصال‌ها نمی‌توانند به طور همزمان استفاده شوند.

هدف شهرداری این است که شبکه‌ای طراحی شود که:

۱. تمام ایستگاه‌ها به یکدیگر متصل باشند
۲. هزینه‌ی کل اتصال‌ها حداقل شود
۳. محدودیت استفاده از انواع کابل‌ها رعایت گردد

ورودی

اطلاعات ورودی شامل موارد زیر است:

- عدد n : تعداد ایستگاه‌ها
- عدد m : تعداد مسیرهای ممکن برای اتصال
- سپس m خط که هر خط اطلاعات یک مسیر را به صورت زیر نشان می‌دهد:

$u \ v \ w \ t$

که در آن:

- u و v : شماره‌ی ایستگاه‌ها ($u, v \leq n$) و ($1 \leq n$)
- w : هزینه‌ی اتصال بین دو ایستگاه

- t : نوع کابل (یک عدد صحیح ≤ 1)

📌 قید مهم

در شبکه‌ی نهایی (درخت پوشای کمینه):

- از هر نوع کابل (t) حداکثر k اتصال می‌توان استفاده کرد.

📦 خروجی

- اگر بتوان شبکه‌ای مطابق شرایط فوق ساخت:

هزینه‌ی کل حداقل شبکه را چاپ کنید.

- در غیر این صورت:

مقدار -1 را چاپ کنید.

🎯 وظایف شما

(الف)

۱. توضیح دهید چرا می‌توان این مسئله را به صورت یک مسئله‌ی درخت پوشای کمینه (MST) مدل‌سازی کرد.

۲. بیان کنید چرا در این مسئله، الگوریتم کروسکال نسبت به الگوریتم پریم انتخاب مناسب‌تری است.

(ب)

۱. توضیح دهید الگوریتم کروسکال چگونه باید تغییر داده شود تا قید حداکثر تعداد استفاده از هر نوع کابل رعایت شود.

۲. بررسی کنید آیا این تغییر، بهینگی الگوریتم را از بین می‌برد یا خیر.
(پاسخ را کوتاه و استدلالی بیان کنید.)

(ج)

تابعی به زبان پایتون بنویسید که:

- ورودی:

n, m, k و لیست یال‌ها

- خروجی:

هزینه‌ی درخت پوشای کمینه یا 1-

شرایط پیاده‌سازی:

- استفاده از Union-Find با Path Compression الزامی است
- یال‌ها باید بر اساس هزینه به صورت صعودی مرتب شوند
- تعداد استفاده از هر نوع کابل کنترل شود
- اگر پس از اجرای الگوریتم، همه‌ی رأس‌ها به هم متصل نشوند، مقدار 1- برگردانده شود

(د)

۱. پیچیدگی زمانی الگوریتم را محاسبه و بیان کنید.

۲. یک مثال ورودی طراحی کنید که:

- بدون در نظر گرفتن قید کابل‌ها، یک MST معتبر داشته باشد
- اما با اعمال قید حداکثر تعداد کابل از هر نوع، هیچ پاسخ معتبری وجود نداشته باشد

نمونه ورودی و خروجی

ورودی:

```
4 6
1 2 10 50 R
1 3 20 70 O
2 3 15 60 O
2 4 25 80 O
3 4 30 90 O
1 4 50 100 R
```


خروجی:

75 210

سوال ۴ :

🗺️ ✨ مدیریت مسیرهای جادویی در سرزمین Algoria

در سرزمین خیالی Algoria، مجموعه‌ای از شهرها وجود دارد که از طریق مسیرهای جادویی به یکدیگر متصل شده‌اند.

هر مسیر دارای هزینه انرژی است و این هزینه‌ها به صورت پویا تغییر می‌کنند؛ به این معنا که ممکن است در طول زمان کاهش یا افزایش یابند.

فرمانروای Algoria قصد دارد یک سیستم هوشمند برای محاسبه‌ی کوتاه‌ترین مسیرها از پایتخت طراحی کند.

با توجه به تغییرات مکرر هزینه‌ی مسیرها، استفاده از ساختارهای داده‌ی معمولی کارایی لازم را ندارد.

از شما—به عنوان معمار الگوریتم‌های سلطنتی—خواسته شده است سیستمی طراحی کنید که با استفاده از هرم فیبوناچی، عملیات مورد نیاز را با بهره‌وری بالا پشتیبانی کند.

داده‌های ورودی

- یک گراف وزن‌دار، بدون جهت و همبند
- تعداد n شهر (رأس) و m مسیر (یال)
- وزن هر یال یک عدد صحیح مثبت
- رأس شماره‌ی 1 به‌عنوان پایتخت **Algoria** در نظر گرفته می‌شود

وظایف شما

(الف)

یک گراف نمونه طراحی کنید که دارای حداقل ۸ رأس و ۱۲ یال باشد، به‌طوری‌که:

- حداقل ۳ مسیر با وزن یکسان در آن وجود داشته باشد
- حداقل یک مسیر وجود داشته باشد که وزن آن در ادامه کاهش یابد

سپس:

- گراف را رسم کرده و وزن یال‌ها را مشخص نمایید
- توضیح دهید چرا این گراف، برای استفاده از **هرم فیبوناچی** نسبت به **هرم دودویی** مناسب‌تر است

(ب)

فرض کنید الگوریتم **Dijkstra** با استفاده از **هرم فیبوناچی** روی گراف طراحی‌شده اجرا می‌شود.

وضعیت هرم فیبوناچی را پس از مراحل زیر رسم و تحلیل کنید:

- درج اولیه‌ی تمام رأس‌ها
- اولین عملیات **Extract-Min**
- حداقل دو عملیات **Decrease-Key**

در ترسیم خود، موارد زیر را به‌وضوح مشخص نمایید:

- لیست ریشه‌ها (**Root List**)

- درجه‌ی هر درخت
 - گره‌های mark شده
-

(ج)

پیچیدگی زمانی اجرای الگوریتم **Dijkstra** را در دو حالت زیر مقایسه کنید (بر حسب n و m):

- استفاده از هرم دودویی
 - استفاده از هرم فیبوناچی
- سپس توضیح دهید چرا در سناریوی Algolia ، عملیات **Decrease-Key** نقش حیاتی و تعیین‌کننده دارد.
-

(د)

در این بخش، یک الگوریتم برای محاسبه‌ی کوتاه‌ترین مسیر از رأس مبدأ در یک گراف وزن‌دار با وزن‌های غیرمنفی طراحی کنید که:

- به صورت صریح از هرم فیبوناچی استفاده کند
 - به شکل یک شبکه‌کد واحد و منسجم نوشته شود
- الگوریتم باید شامل موارد زیر باشد:
- منطق کامل الگوریتم **Dijkstra**
 - استفاده‌ی واضح از هرم فیبوناچی به عنوان صف اولویت
 - پیاده سازی انتزاعی عملیات‌های:

○ Insert

○ Extract-Min

○ Decrease-Key

📌 نکات مهم:

- نیازی به پیاده‌سازی تمام جزئیات داخلی هرم فیبوناچی نیست

- اما ساختار کلی گره‌ها (parent ، child ، degree ، mark) و نقش آن‌ها در عملیات‌ها باید مشخص باشد
- شبه‌کد باید نشان دهد که عملیات‌های **Decrease-Key** و **Extract-Min** دقیقاً چگونه در اجرای **Dijkstra** فراخوانی می‌شوند
- پاسخ باید فقط شامل یک شبه‌کد واحد باشد
(نوشتن شبه‌کدهای جداگانه برای Dijkstra و Fibonacci Heap مجاز نیست)

(۵)

اگر وزن یال‌ها منفی شوند، اما چرخه‌ی منفی وجود نداشته باشد:

- آیا همچنان استفاده از هرم فیبوناچی منطقی است؟
- در این شرایط، کدام الگوریتم را جایگزین می‌کنید و چرا؟

نمونه ورودی و خروجی

ورودی:

4 5
2 8
1 3
1 4
1 5
1 2
1 3
1 4

خروجی:

12

