
سوال ۱ :

 محاسبه کارایی یک دنباله عددی

در یک سامانه نرم‌افزاری که برای انجام محاسبات تکراری روی دنباله‌های عددی استفاده می‌شود، مشاهده شده است که با بزرگ شدن اندازه ورودی، زمان اجرای برنامه به طور قابل توجهی افزایش می‌یابد. هدف این است که این بخش از سامانه به گونه‌ای پیاده‌سازی شود که برای ورودی‌های بزرگ نیز در زمان قابل قبول اجرا شود. دنباله موردنظر به صورت زیر تعریف شده است:

$$F(0) = 0, F(1) = 1, F(n) = F(n-1) + F(n-2) \text{ برای } n \geq 2$$

 وظیفه شما

برنامه‌ای بنویسید که با دریافت یک ورودی، مقدار $F(n)$ را محاسبه کند.

 ورودی

یک عدد صحیح غیرمنفی n در بازه $0 \leq n \leq 10^6$ که نشان‌دهنده اندیس دنباله است. (توجه: ممکن است ورودی نامعتبر یا خارج از بازه باشد.)

- در صورت معتبر بودن ورودی، یک عدد صحیح که برابر با $F(n) \bmod (10^9 + 7)$ است.
- در صورت نامعتبر بودن ورودی (عدد صحیح نباشد یا خارج از بازه مجاز باشد)، عبارت INVALID چاپ شود.

قوانین مسئله

- مقدار n در بازه 0 تا 10^6 (شامل هر دو) معتبر است.
- به دلیل بزرگ شدن سریع اعداد، خروجی باید به پیمانه $10^9 + 7$ محاسبه شود.
- برنامه باید برای n های بزرگ (حتی نزدیک 10^6) در زمان مناسب اجرا شود (راهکارهایی مانند ذخیره‌سازی مقادیر محاسبه‌شده یا روش‌های بهینه پیشنهاد می‌شود).
- اگر ورودی عدد صحیح نباشد یا خارج از بازه باشد، خروجی INVALID است.

نمونه ورودی و خروجی

ورودی:

5
-3

خروجی:

5

INVALID

سوال ۲ :

🏴‍☠️💰 |بیشینه‌سازی ارزش گنج با محدودیت شباهت و ترتیب صعودی

شما یک دزد دریایی حرفه‌ای هستید و یک نقشه گنج اسرارآمیز پیدا کرده‌اید. نقشه شامل یک رشته حروف بزرگ انگلیسی است که محل‌های گنج ارزشمند را نشان می‌دهد.

در جزیره، n صندوق گنج وجود دارد که هر صندوق شامل:

- کد مخفی صندوق (رشته‌ای از حروف بزرگ انگلیسی)
- ارزش طلاهای داخل آن (عدد صحیح مثبت)

شرایط انتخاب صندوق‌ها:

۱. شما می‌توانید فقط صندوق‌هایی را انتخاب کنید که کد آن‌ها حداقل K حرف با نقشه شباهت داشته باشد.
شباهت به صورت بیشترین زیررشته مشترک بین کد صندوق و نقشه محاسبه می‌شود.
۲. صندوق‌ها باید به ترتیبی انتخاب شوند که ارزش طلاها هرگز کاهش پیدا نکند (غیرنزولی باشد).
۳. هدف شما بیشینه کردن مجموع ارزش طلاهای انتخاب شده است.

🎯 وظیفه شما

برنامه‌ای طراحی کنید که با دریافت اطلاعات صندوق‌ها و نقشه، بیشترین مجموع ارزش طلاها را با رعایت شرایط فوق محاسبه کند.

n K

R

S_1 v_1

S_2 v_2

...

S_n v_n

• خط اول:

- n : تعداد صندوق ها ($1 \leq n \leq 500$)
- K : حداقل میزان شباهت لازم با نقشه ($1 \leq K \leq 100$)

• خط دوم:

- R : رشته نقشه گنج ($1 \leq \text{Len}(R) \leq 200$) و فقط حروف بزرگ انگلیسی

• خطوط بعدی:

- S_i : رشته کد صندوق ($1 \leq \text{Len}(s_i) \leq 200$) و فقط حروف بزرگ

انگلیسی

- V_i : ارزش طلاهای صندوق ($1 \leq v_i \leq 10^9$)

نکته مهم: فقط فاصله (space) بین رشته و عدد معتبر است. ورودی با کاما یا جداکننده غیر از فاصله نامعتبر است.

- اگر ورودی معتبر باشد: یک عدد صحیح که نشان‌دهنده بیشترین مجموع ارزش طلاها با رعایت شرایط است.
- اگر هیچ صندوق معتبری ($K \geq$ با شباهت) وجود نداشته باشد -> خروجی 0
- اگر ورودی نامعتبر باشد (فرمت اشتباه، کاراکتر غیرمجاز، جداکننده نامعتبر، اعداد خارج از محدوده، رشته‌های خالی یا شامل حروف غیر بزرگ انگلیسی):
INVALID INPUT خروجی

📌 قوانین مسئله

- شباهت = طول بیشترین زیررشته مشترک (Longest Common Substring) بین کد صندوق و نقشه.
- انتخاب صندوق‌ها می‌تواند هر زیردنباله‌ای (نه لزوماً متوالی) از صندوق‌ها باشد، به شرطی که ارزش‌ها غیرنزولی باشند.
- در پیاده‌سازی نهایی، ورودی‌ها ابتدا به طور کامل خوانده شده و سپس اعتبارسنجی می‌شوند تا در محیط‌های تعاملی از خطاهای ناشی از ورود ناقص داده جلوگیری شود.
- محدودیت‌ها:
 - $n \leq 500$ ← می‌توان از $O(n^2)$ یا بهتر استفاده کرد.
 - طول رشته‌ها ≥ 200 ← محاسبه LCS (زیررشته مشترک) با $O(200^2)$ (2002) برای هر صندوق مشکلی ندارد.

نمونه ورودی و خروجی

ورودی:

5 2

ABCDEF

AB 10

AC 20

BC 15

DEF 25

XYZ 30

خروجی:

55

سوال ۳ :

 شبکه نجات جزایر در طوفان

کشور شما از چندین جزیره کوچک و بزرگ تشکیل شده است. اخیراً طوفان‌های شدید باعث شده برخی جزایر از هم جدا شوند و دسترسی به مراکز امداد و نجات محدود شود.

دولت تصمیم گرفته یک شبکه پل‌های اضطراری و مسیرهای ارتباطی سریع و مقاوم بسازد تا:

- همه جزایر به یکدیگر متصل باشند.
- مسیرها ایمن باشند و بتوانند در برابر طوفان و شرایط بحرانی مقاومت کنند.
- هزینه ساخت و نگهداری مسیرها تا حد امکان کم باشد.

اما نکات چالش برانگیز وجود دارد:

- هر مسیر بین دو جزیره دارای هزینه ساخت w (طول پل، تجهیزات، سختی مسیر) s و استحکام (میزان مقاومت در برابر شرایط بحرانی، عدد بین ۱ تا ۱۰۰)
- برخی مسیرها طولانی اما مقاوم هستند، برخی کوتاه اما ضعیف.
- پل‌های بحرانی: برخی مسیرها در مناطق پرخطر قرار دارند. استفاده از آن‌ها می‌تواند مجموع استحکام MST را افزایش دهد اما هزینه آن‌ها بالاتر است.

 وظیفه شما به عنوان مهندس نجات

۱. ایجاد شبکه پایه با حداقل هزینه:

شبکه‌ای بسازید که همه جزایر متصل باشند. (MST) از الگوریتم پریم (Prim) برای ساخت MST استفاده کنید.

۲. بهینه‌سازی استحکام در صورت تساوی هزینه:

اگر هنگام انتخاب مسیر در پریم چند مسیر با هزینه یکسان وجود داشته باشد،

مسیر با استحکام بالاتر انتخاب شود.
هدف MST: با کمترین هزینه و بیشترین مجموع استحکام.

۳. مدیریت مسیرهای بحرانی و محدودیت محیطی:
برخی مسیرها در مناطق خطرناک هستند. انتخاب آنها باعث افزایش استحکام کل MST می‌شود.
دانشجو باید هوشمندانه مسیرها را اولویت‌بندی کند تا MST نهایی هم کم‌هزینه و هم مقاوم باشد.

۴. چالش اضافه – پل‌های اجباری و اختیاری:
در ورودی، برخی مسیرها با حرف R علامت‌گذاری می‌شوند (Required) که حتماً باید در MST باشند.
مسیرهای بدون R (با علامت O – Optional) اختیاری هستند و پریم باید انتخاب کند که کمترین هزینه و بیشترین استحکام را بدهد.

۵. ملاحظات عملیاتی:

- اگر MST ممکن نباشد (جزیره‌ای جدا از بقیه باشد یا مسیرهای اجباری باعث ایجاد دور شوند)، خروجی INVALID INPUT باشد.
- ورودی نامعتبر شامل:
 - شماره جزایر خارج از محدوده
 - هزینه یا استحکام غیرمجاز
 - مسیرهای موازی یا مسیرهای خود-اتصال ($u = v$)
 - پرچم‌های نامعتبر (نه R و نه O)

n m

u_1 v_1 w_1 s_1 $flag_1$

u_2 v_2 w_2 s_2 $flag_2$

...

u_m v_m w_m s_m $flag_m$

• خط اول:

○ n : تعداد جزایر ($1 \leq n \leq 500$)

○ m : تعداد مسیرها ($1 \leq m \leq \frac{n(n-1)}{2}$)

• خطوط بعدی: هر خط شامل 5 مورد با جداکننده فاصله (space)

○ u, v : شماره جزایر متصل ($1 \leq u, v \leq n, u \neq v$)

○ w : هزینه ساخت مسیر ($1 \leq w \leq 10^6$)

○ S : استحکام مسیر ($1 \leq s \leq 100$)

○ $flag$: R برای مسیرهای اجباری، 0 برای مسیرهای اختیاری

تضمین: هیچ مسیر موازی یا مسیر خود-اتصال وجود ندارد ($u = v$).

- اگر ورودی نامعتبر باشد INVALID INPUT :
 - در غیر این صورت : دو عدد صحیح جدا شده با فاصله:
هزینه_کل مجموع_استحکام
 - هزینه کل : مجموع هزینه مسیرهای MST با کمترین هزینه ممکن و شامل تمام مسیرهای اجباری
 - مجموع استحکام : مجموع استحکام مسیرهای MST با بیشترین مجموع استحکام ممکن در صورت تساوی هزینه
-

قوانین مسئله

- الگوریتم پایه : پریم (Prim) با پیچیدگی $O(m \log n)$ یا $O(n^2)$ مناسب است.
 - اولویت انتخاب یال در پریم :
 ۱. هزینه کمتر ارجح است.
 ۲. در صورت تساوی هزینه، استحکام بیشتر ارجح است.
 - مسیرهای اجباری (R) باید حتماً در MST نهایی حضور داشته باشند.
 - اگر مسیرهای اجباری باعث ایجاد دور شوند یا گراف را ناهمبند کنند ← INVALID INPUT .
 - اگر گراف اصلی ناهمبند باشد ← INVALID INPUT .
 - اعتبارسنجی ورودی باید ابتدا انجام شود (اعداد در محدوده، پرچم‌ها معتبر، بدون مسیر تکراری یا خود-اتصال).
-

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

☁️⚡ مدیریت سرورهای حیاتی در شبکه توزیع شده ابری

یک شرکت بین‌المللی ارائه‌دهنده‌ی خدمات ابری (Cloud Services) در حال مدیریت یک شبکه‌ی توزیع‌شده از مراکز داده (Data Centers) است. به دلیل توسعه‌ی تدریجی زیرساخت‌ها، ساختار ارتباط بین مراکز داده به‌صورت یک درخت ارتباطی شکل گرفته است؛ به این معنی که شبکه بدون حلقه بوده و بین هر دو مرکز داده دقیقاً یک مسیر یکتا وجود دارد.

هر مرکز داده دارای تعدادی سرور حیاتی است که در این مسئله، هر مرکز داده را به‌صورت یک (گره) در نظر می‌گیریم. فعال بودن هر گره دارای هزینه‌ی انرژی و یک میزان اهمیت عملیاتی است که نشان می‌دهد فعال بودن آن مرکز داده چه میزان به پایداری، سرعت پاسخ‌گویی و کیفیت سرویس شبکه کمک می‌کند.

با توجه به بحران انرژی و محدودیت‌های فنی سیستم‌های خنک‌کننده، شرکت تنها مقدار محدودی انرژی در اختیار دارد. از طرفی، به دلیل محدودیت‌های زیرساخت شبکه و جلوگیری از اضافه‌بار مسیرهای ارتباطی، هیچ دو مرکز داده‌ی مستقیماً متصل به یکدیگر نباید به‌صورت هم‌زمان فعال باشند؛ چرا که فعال بودن هم‌زمان آن‌ها باعث ازدحام ترافیک، افزایش تأخیر و بروز ناپایداری در شبکه می‌شود.

مدیر فنی شبکه می‌خواهد بداند کدام مراکز داده را باید فعال نگه دارد تا:

- مجموع انرژی مصرفی از حد مجاز فراتر نرود،
- هیچ دو مرکز داده‌ی همسایه هم‌زمان فعال نباشند،
- و مجموع اهمیت عملیاتی مراکز فعال حداکثر ممکن باشد.

برنامه‌ای طراحی کنید که با دریافت اطلاعات مراکز داده و ساختار درخت ارتباطی، بیشترین مجموع اهمیت عملیاتی قابل دستیابی را با رعایت محدودیت‌های انرژی و مجاورت محاسبه کند.

ورودی 📁

n C
c1 v1
c2 v2
...
cn vn
u1 v1
u2 v2
...
u(n-1) v(n-1)

• خط اول:

- n : تعداد مراکز داده (تعداد گره‌ها، $1 \leq n \leq 200$)
- C : حداکثر انرژی قابل مصرف (عدد صحیح، $1 \leq C \leq 10000$)

• خطوط 2 تا n+1 (n از 1 تا i برای هر گره)

- c_i : مصرف انرژی مرکز داده‌ی i ام (عدد صحیح، $c_i \leq C \leq 1$)
- v_i : میزان اهمیت عملیاتی مرکز داده‌ی i ام (عدد صحیح، $v_i \leq 1 \leq 10^9$)

• خطوط بعدی (خط $n-1$)

○ $u_i v_i$: وجود ارتباط مستقیم بین مراکز داده‌ی u_i و v_i (یال‌های درخت)

نکته :گره‌ها از ۱ تا n شماره‌گذاری شده‌اند.

تضمین :شبکه ورودی حتماً باید یک درخت معتبر باشد (یال، بدون حلقه ، همبند $n-1$)

خروجی

- اگر ورودی معتبر باشد: یک عدد صحیح که نشان‌دهنده بیشترین مجموع اهمیت عملیاتی قابل دستیابی است.
- اگر ورودی نامعتبر باشد(شامل: اعداد خارج از محدوده، مصرف انرژی خارج از بازه، عدم وجود $n-1$ یال، وجود حلقه، ناهمبندی، فرمت نامناسب، تکرار یال، یال خود-اتصال) : خروجی INVALID

📌 قوانین مسئله

- محدودیت مجاورت :اگر دو گره u و v مستقیماً به هم متصل باشند (یال بین آنها وجود داشته باشد)، نمی‌توانند هر دو همزمان فعال باشند.
 - محدودیت انرژی :مجموع انرژی مصرفی گره‌های فعال $\leq C$.
 - هدف :پیشینه کردن مجموع اهمیت عملیاتی گره‌های فعال.
-

نمونه ورودی و خروجی

ورودی:

4 5

2 8

1 3

1 4

1 5

1 2

1 3

1 4

خروجی:

12