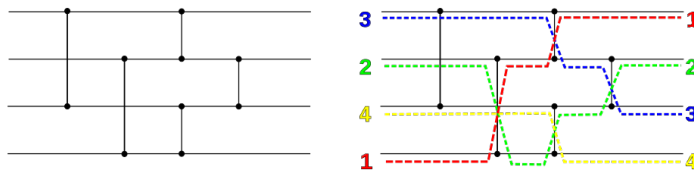


شبکه‌های مرتب‌سازی

شبکه‌ی مرتب‌سازی^۱ یک مدل انتزاعی ریاضی شامل شبکه‌ای از سیم‌ها و واحدهای مقایسه کننده است که برای مرتب سازی دنباله‌ای از اعداد از آن استفاده می شود. هر مقایسه کننده دو سیم را به هم متصل می کند و مقادیر را با قرار دادن مقدار کوچکتر روی یکی از سیم‌ها و مقدار بزرگتر روی سیم دیگر، مرتب می کند. هر سیم دارای یک مقدار می باشد، و هر مقایسه کننده دو سیم را به عنوان ورودی و خروجی می گیرد. زمانی که دو مقدار وارد مقایسه کننده می شود، مقایسه کننده مقدار کوچکتر را در سیم بالاتر، و مقدار بزرگتر را در سیم پایین قرار می دهد. به شبکه‌ای از سیم‌ها و مقایسه کننده‌ها که به طور صحیح تمام مقادیر ورودی را به صورت صعودی مرتب کنند یک شبکه مرتب‌سازی گفته می شود.

شکل زیر (چپ)، نشان دهنده‌ی یک شبکه‌ی مرتب‌سازی است. سیم‌ها در این شکل به صورت افقی و مقایسه کننده‌ها به صورت عمودی نشان داده شده‌اند. مراحل انجام مرتب‌سازی توسط این شبکه در شکل راست نشان داده شده است. فهم چگونگی صحیح عمل کردن این شبکه مرتب‌سازی آسان است. این را مدنظر داشته باشید که مقایسه کننده‌ها مقدار بزرگ را به سیم پایین و مقدار کوچک را به سیم بالا منتقل می کنند.



هدف این تمرین این است که عملکرد یک شبکه را روی یک دنباله‌ی ورودی شبیه‌سازی کنیم. در ورودی، شبکه و دنباله‌ی اعداد داده می شوند. خروجی برنامه تعیین می کند که آیا شبکه‌ی داده شده اعداد را به درستی مرتب می کند یا نه. توضیح این که لزوماً هر شبکه، به طور صحیح اعداد را مرتب نمی کند. مثال‌هایی از حالت‌های درست و نادرست در نمونه‌های زیر ذکر می شوند. علاوه بر این، ممکن است در توصیف شبکه‌ی ورودی خطاهایی وجود داشته باشد که برنامه‌ی شما باید آنها را تشخیص دهد. برای توصیف یک شبکه‌ی ورودی، از ماتریسی از کاراکترها استفاده می کنیم. به عنوان مثال شبکه‌ی نشان داده شده در شکل فوق توسط ماتریس زیر توصیف می شود.

a-c-
-bce
a-de
-bd-

هر سطر از ماتریس یک سیم را مشخص می کند. هر کاراکتر از یک سطر یا '-' است که نشان دهنده‌ی عدم وجود مقایسه کننده در آن بخش است یا با یک حرف لاتین مشخص می شود که در این صورت، باید در یکی (و تنها یکی) دیگر از کاراکترهای آن ستون کاراکتر مشابهی پیدا شود. به این ترتیب، فرض می شود یک واحد مقایسه کننده بین سیم‌های متناظر آن دو سطر وجود دارد. به عنوان مثال، در توصیف فوق یک مقایسه کننده بین دو سیم اول و سوم وجود دارد که با حرف a مشخص شده. ستون دوم نیز وجود یک مقایسه کننده بین سیم‌های دوم و چهارم است. ترتیب انتقال اعداد از چپ به راست فرض می شود.

در توصیف داده شده از شبکه ممکن است خطاهایی به شرح زیر وجود داشته باشد که برنامه‌ی شما باید آنها را تشخیص دهد:

- در یک ستون تنها یک مورد از یک حرف پیدا می شود یا این که بیش از دو مورد از آن حرف وجود دارد.
- در توصیف شبکه کاراکتری غیر از حروف کوچک لاتین و '-' وجود دارد.

دقت کنید که دو مقایسه کننده در دو ستون مختلف می توانند با یک حرف یکسان نمایش داده شوند، چون این امر ابهامی در عملکرد شبکه ایجاد نمی کند.

ورودی

خط اول هر مورد آزمون حاوی دو عدد صحیح N و K است که به ترتیب تعداد سطرها و تعداد ستون‌های ماتریس شبکه را مشخص می کنند. بعد از آن، N خط پشت سر هم می آیند که هر یک از K کاراکتر تشکیل شده‌اند و محتوای کاراکترهای ماتریس را مشخص می کنند. سپس در یک خط، N عدد صحیح که به ترتیب روی سیم‌های ۱ تا N قرار خواهند گرفت ذکر می شوند. در ورودی مسئله ممکن است تعداد بیش از یک مورد آزمون ذکر شود که هر یک از قالب فوق پیروی می کند. آخرین خط ورودی، شامل دو عدد صفر است.

^۱ sorting network

خروجی

برای هر مورد آزمون، یک خط در خروجی بنویسید که با قالبی مانند آنچه در بخش «نمونه خروجی» آمده، یکی از سه نتیجه‌ی ممکن را مشخص کند: **Not sorted** به این معنی که اعداد ورودی پس از پردازش توسط شبکه مرتب نمی‌شوند، **Sorted** به این معنی که اعداد ورودی پس از پردازش توسط شبکه مرتب می‌شوند و **Invalid network** به این معنی که ورودی داده‌شده قوانین مطرح‌شده در توصیف مسئله را نقض می‌کند.

نمونه خروجی

```
Sorted
Not_sorted
Invalid_network
```

نمونه ورودی

```
4 4
ac-e
a-de
b-df
bc-f
1 3 2 0
3 2
a-
ab
-b
1 3 0
3 2
a-
ab
ab
1 3 0
0 0
```

نحوه‌ی تحویل

اگر شماره‌ی دانشجویی شما 810189123 است، نام فایل برنامه‌ی خود را `A1-89123.cpp` بگذارید و آن را در محل مربوطه در CECM قرار دهید. پیش از تحویل مطمئن شوید برنامه‌ی شما قابل ترجمه است.

دقت کنید

- برنامه‌ی خود را فقط در قالب یک فایل به زبان سی‌پلاس‌پلاس تحویل دهید.
- به جز کتابخانه‌های استاندارد زبان سی‌پلاس‌پلاس و `std.lib.facilities.h` از هدر فایل دیگری استفاده نکنید.
- قبل از تحویل، برنامه‌ی خود را با انواع ورودی‌ها بیازمایید. حالت‌های مرزی ورودی‌های مسئله را در نظر بگیرید.
- به قالب خروجی دقت کنید. بزرگی و کوچکی حروف، فاصله‌های خالی و شکست خط‌ها مهم هستند. به هیچ‌وجه چیزی به جز آنچه در بخش «خروجی» اشاره شده در خروجی برنامه ننویسید (مثل پیغام‌هایی به کاربر برنامه).

مطالعه‌ی بیشتر

آیا می‌دانید قضیه‌ای وجود دارد که اگر شبکه‌ای هر دنباله‌ی ورودی از صفر و یک را به درستی مرتب کند، تمام دنباله‌های دیگر از اعداد دلخواه را نیز به درستی مرتب می‌کند؟ شبکه‌های مرتب‌سازی در عین سادگی دارای مباحث نظری عمیق و جالبی هستند. برای مطالعه‌ی بیشتر می‌توانید به فصل ۲۷ کتاب زیر مراجعه کنید.

T. Cormen, C. Leiserson, R. Rivest, and C. Stein, "Introduction to Algorithms", 2nd edition, MIT Press, 2001.

همچنین صفحه‌ی **Sorting Networks** از ویکی‌پدیا نیز دارای پیوندهای مفیدی در این زمینه است. بخشی از توضیحات این بخش از ویکی‌پدیای فارسی برداشته شده است.