۱۰ نمره ....

رستورانی را در نظر بگیرید که دارای ۲۳ صندلی با شماره‌های ۱ تا ۲۳ است. این صندلی‌ها در یک خط مستقیم قرار دارند. فرض کنید که مشتریان این رستوران، به صورت یک‌نفره و یا در دسته‌های دونفره وارد رستوران می‌شوند و اعضای هر دسته‌ی دونفره با هم از رستوران خارج می‌شوند. همچنین فرض کنید که هیچ‌گاه در یک زمان بیشتر از ۱۶ نفر مشتری در این رستوران وجود ندارد.

ثابت کنید که اگر هیچ مشتری یک‌نفره در صندلی‌های با شماره‌ی ۲، ۵، ۸، ۱۱، ۱۴، ۱۷ و ۲۰ ننشینند، آنگاه همواره می‌توان مشتری‌های دونفره را بدون جدا کردن از یکدیگر در صندلی‌های کنار هم در رستوران نشاند. (توجه داشته باشید که هیچ مشتری نشسته را نمی‌توان تغییر مکان داد.)

۱۰ نمره ....

در کارخانه‌ای یک دستگاه وجود دارد که باید n کار را انجام دهد. می‌دانیم که انجام کار i ام به اندازه‌ی t_i از این دستگاه وقت می‌گیرد و باید حداکثر تا زمان d_i تحویل داده شود. فرض کنید که دستگاه در زمان صفر شروع به کار می‌کند. علاوه بر این، می‌دانیم که این دستگاه نمی‌تواند در هر لحظه بیش از یک کار را انجام دهد.

اگر دستگاه در زمان s_i شروع به انجام کار i ام کند، انجام آن در زمان $s_i + t_i$ به پایان خواهد رسید. اگر $s_i + t_i > d_i$، یعنی کار i ام در زمانی که باید تحویل داده شود هنوز به طور کامل انجام نشده باشد، مقدار $L_i = s_i + t_i - d_i$ را دیرکرد کار i ام می‌نامیم. در غیر این صورت دیرکرد کار i ام برابر با صفر تعریف می‌شود. دیرکرد کل دستگاه برابر با بیشترین دیرکرد کارها، یعنی $L = \max\{L_۱, L_۲, \ldots, L_n\}$ تعریف می‌شود.

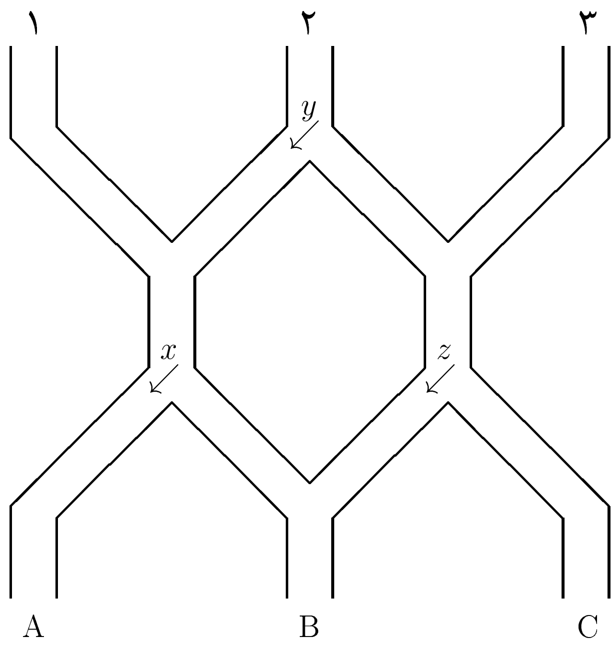
می‌خواهیم ترتیب انجام کارها را به گونه‌ای پیدا کنیم که مقدار دیرکرد کل دستگاه حداقل شود. برای این منظور الگوریتمی به این صورت پیشنهاد داده شده است:

ابتدا کارها را برحسب مقدار d_i آنها به ترتیب صعودی مرتب می‌کنیم و دستگاه کارها را به این ترتیب انجام می‌دهد.

ثابت کنید که این الگوریتم درست عمل می‌کند، یعنی اگر کارها را به این ترتیب انجام دهیم، مقدار دیرکرد کل دستگاه حداقل می‌شود.

.... **۱۵ نمره**

دستگاهی مانند شکل زیر را در نظر بگیرید:



در هر یک از ورودی‌های ۱، ۲ و ۳ می‌توانیم یک گلوله بیندازیم. این گلوله به سوی پایین حرکت می‌کند و با توجه به وضعیت کلیدهای x، y و z از یکی از خروجی‌های A، B یا C خارج می‌شود. کلیدهای x، y و z به این صورت عمل می‌کنند: هر کلید می‌تواند در یکی از دو وضعیت $\searrow$ یا $\swarrow$ باشد. اگر کلید در وضعیت $\searrow$ باشد، گلوله را به سمت راست و اگر در وضعیت $\swarrow$ باشد، گلوله را به سمت چپ می‌فرستد. علاوه بر این با عبور هر گلوله از یک کلید، وضعیت آن کلید تغییر می‌کند.

در ابتدای شروع کار دستگاه، هر سه کلید در وضعیت $\swarrow$ هستند. یک دنباله مانند $a_1 a_2 \cdots a_n$، ($a_i \in \{1, 2, 3\}$ برای هر i) به عنوان دنباله‌ی ورودی دستگاه داده می‌شود. پس از این ابتدا یک گلوله از ورودی شماره‌ی a_1، سپس یک گلوله از ورودی شماره‌ی a_2، ... و در انتها یک گلوله از ورودی شماره‌ی a_n به درون دستگاه می‌اندازیم. فرض می‌کنیم که گلوله‌ها به ترتیب از خروجی‌های b_1، b_2، ... و b_n خارج شوند ($b_i \in \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$ برای هر i). دنباله‌ی $b_1 b_2 \cdots b_n$ را دنباله‌ی خروجی دستگاه برای ورودی $a_1 a_2 \cdots a_n$ می‌نامیم.

به عنوان مثال دنباله‌ی خروجی دستگاه برای ورودی ۱۲۳۲۱، دنباله‌ی ABBCA است.

الف) الگوریتمی بنویسید که با دریافت یک دنباله‌ی ورودی، دنباله‌ی خروجی آن را پیدا

کند.

ب) الگوریتمی بنویسید که با دریافت یک دنباله‌ی $s_1 s_2 \cdots s_n$ ($s_i \in \{\mathrm{A}, \mathrm{B}, \mathrm{C}\}$ برای هر i) مشخص کند که آیا این دنباله می‌تواند خروجی دستگاه باشد یا خیر؟ الگوریتم شما باید سریع باشد، یعنی امتحان کردن تمام حالتها مورد نظر نیست.

۱۵ نمره ....

یک دسته کارت شامل $2n$ کارت که روی آنها عددهای $0, 1, \ldots, 2n-1$ نوشته شده است، داده شده است. می‌توانیم با انجام عمل زیر روی این دسته کارت، یک دسته کارت دیگر که در آن ترتیب قرار گرفتن کارتها تغییر کرده‌است، بسازیم:

ابتدا دسته کارت را به دو دسته که اولی شامل n کارت اول و دومی شامل n کارت باقیمانده است، تقسیم می‌کنیم. سپس به ترتیب یک کارت از دسته‌ی اول و یک کارت از دسته‌ی دوم برمی‌داریم و این کار را آن قدر تکرار می‌کنیم تا تمام کارتها برداشته شوند.

به عنوان مثال اگر شماره‌ی کارتهای قرار گرفته در دسته‌ی اول به ترتیب برابر با ۷, ۱, ۶, ۲, ۵, ۴, ۳, ۸ باشد، پس از انجام عمل فوق، ترتیب قرار گرفتن کارتها به صورت ۷, ۵, ۱, ۴, ۶, ۳, ۲, ۸ خواهد بود.

عمل فوق را «» دسته کارت می‌نامیم.

الف) ثابت کنید که برای هر n، اگر دسته کارت را بُر بزنیم، سپس دسته کارت حاصل را دوباره بُر بزنیم و همین کار را تکرار کنیم، بالاخره پس از مدتی به همان دسته کارت اولیه می‌رسیم.

ب) برای $n = 10$ چند بار باید عمل بُر زدن را تکرار کنیم تا به دسته کارت اولیه برسیم؟ (برای جواب خود دلیل بیاورید.)

ج) ثابت کنید که برای $n = 2^k$ پس از $k+1$ بار بُر زدن به دسته کارت اولیه می‌رسیم.

د) ثابت کنید که برای $n = 2^k+1$ پس از $2k+2$ بار بُر زدن به دسته کارت اولیه می‌رسیم.