جمهوری اسلامی ایران
وزارت آموزش و پرورش

دفترچه سوالات

مرحله دوم

هفتمین دوره‌ی المپیاد کامپیوتر سال ۱۳۷۵

مدت آزمون (دقیقه)	تعداد سوالات	
	مساله‌های تشریحی	سوالات چند گزینه‌ای
۳۰۰	۸	-

توضیحات مهم

استفاده از ماشین حساب مجاز نیست.

۱- جابه‌جایی مهره‌ها

یک صفحه‌ی 3×3 را در نظر بگیرید که دو مهره‌ی سفید و دو مهره‌ی سیاه در آن به‌صورت شکل «الف» قرارگرفته‌اند. B نشان‌دهنده‌ی مهره‌ی سیاه و W نشان‌دهنده‌ی مهره‌ی سفید است.

W		B
W		B

شکل «الف»

B		W
W		B

شکل «ب»

هر یک از مهره‌ها را می‌توان به این صورت حرکت داد: آن مهره را برداشته و در خانه‌ای که دو ستون و یک سطر، یا دو سر و یک ستون با آن فاصله دارد قرار می‌دهیم؛ مشروط بر این‌که خانه‌ی مقصد خالی باشد. (این نوع حرکت همان حرکت مهره‌ی اسب در شطرنج است).

آیا می‌توانیم با حرکت دادن مهره‌ها، به تعداد و ترتیب دلخواه، موقعیت صفحه را به شکل «ب» تبدیل کنیم؟ توضیح دهید.

۲- بازی سنگریزه‌ها

دو دسته سنگریزه که در یکی از آن‌ها m و در دیگری n سنگریزه قرار دارد در نظر بگیرید. دو بازیکن زیر را با این سنگریزه‌ها انجام می‌دهند:

هر بازیکن در نوبت خود، از یکی از دسته‌ها (یک دسته‌ی دلخواه که حداقل دو سنگریزه داشته باشد) دو سنگریزه برداشته و یکی از آن‌ها را به دسته‌ی دیگر اضافه می‌کند. دو بازیکن یکی در میان این حرکت را انجام می‌دهند تا جایی که دیگر حرکتی امکان نداشته باشد. در این هنگام کسی که آخرین حرکت را انجام داده است، برنده‌ی بازی محسوب می‌شود.

شرط لازم و کافی برای m و n را به‌دقت آورید که نفر دوم بتواند طوری بازی کند که برنده‌ی بازی شود.

۳- خروجی الگوریتم

عمل $\oplus$ به‌صورت زیر تعریف می‌شود:

فرض کنید که نمایش عددهای x و Y در مبنای دو به‌صورت $x = x_nx_{n-1}\dots x_1x_0$ و $y = y_ny_{n-1}\dots y_1y_0$ باشد. (در صورت لزوم در سمت چپ نمایش دودویی عدد کوچک‌تر به تعداد موردنظر صفر اضافه می‌کنیم.) برای هر i $(0 \le i \le n)$؛درصورتی‌که دقیقاً یکی از دو عدد x_i و y_i برابر با یک و دیگری صفر باشد، a_i را مساوی با یک و در غیر این صورت مساوی با صفر تعریف می‌کنیم. عددی که نمایش آن در مبنای دو به‌صورت $a_na_{n-1}\dots a_1a_0$ است برابر با $x\oplus y$ خواهد بود. حال الگوریتم زیر را در نظر بگیرید:

۱) a_0 را مساوی با ۱ و k را مساوی با ۱ قرار بده.

۲) a_k را مساوی با a_{k-1} قرار بده.

۳) به مقدار a_k یکی اضافه کن.

٤) F را برابر ۱ قرار بده.

٥) برای هر i از صفر تا $k-1$، $(0 \le i < k)$ این کار را انجام بده:

۵-۱) برای هر j از صفر تا $k-۱$، $(۰ \leq j < k)$ این کار را انجام بده:

۵-۱-۱) درصورتی‌که $a_k = a_i \oplus a_j$ است، F را مساوی با ۰ قرار بده.

٦) اگر $F = ۱$ است. به مقدار k یکی اضافه کن و در غیر این صورت به مرحله‌ی ۳ برو.

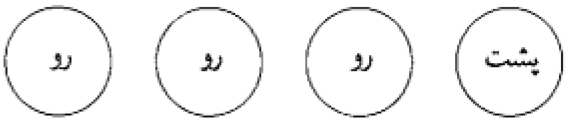
۷) اگر $k \leq ۱۳۷۶$ است، به مرحله‌ی (۲) برو و در غیر این صورت متوقف شو.

مقدار $a_{۱۳۷۶}$ در انتهای این الگوریتم چند است؟ برای ادعای خود دلیل بیاورید.

٤- پشت‌ورو کردن سکه‌ها

n سکه در یک ردیف در کنار هم قرار دارند. بعضی از این سکه‌ها به رو و بعضی به پشت قرارگرفته‌اند. در هر حرکت می‌توانیم یکی از این n سکه را انتخاب کنیم و آن سکه و سکه‌های مجاور سمت راست و سمت چپ آن را هم‌زمان برگردانیم (از رو به پشت یا از پشت به رو). توجه کنید که درصورتی‌که سکه‌ی انتخاب‌شده یکی از دو سکه‌ی انتهایی باشد، دو سکه و در غیر این صورت سه سکه برگردانده می‌شود.

الف) ثابت کنید که اگر $n = ۳k$ یا $n = ۳k + ۱$ باشد، به هر ترتیبی که سکه‌ها قرارگرفته باشند، با استفاده از چنین حرکت‌هایی می‌توانیم همه‌ی سکه‌ها را به رو برگردانیم. برای مثال اگر $n = ۴$ و ترتیب اولیه‌ی سکه‌ها به‌صورت زیر باشد:



می‌توانیم اول سکه‌ی دوم (از سمت چپ)، سپس سکه‌ی اول و نهایتاً سکه‌ی چهارم را انتخاب کنیم تا همه‌ی سکه‌ها به رو برگردانده شوند.

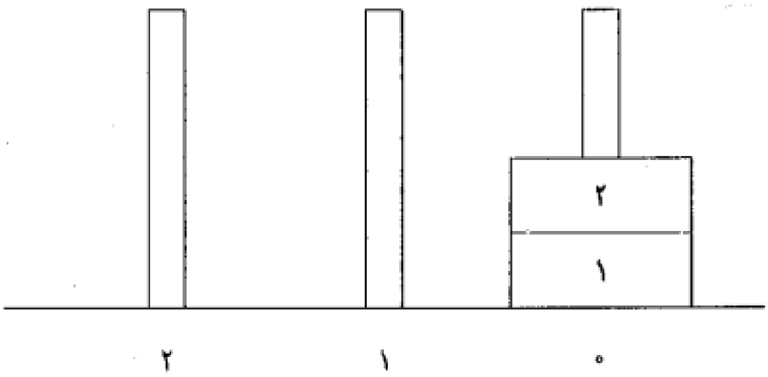
ب) ثابت کنید که برای هر nکه به‌صورت $n = ۳k + ۲$ باشد، وضعیت اولیه‌ای وجود دارد که برای آن با استفاده از این حرکت‌ها نمی‌توان این کار را انجام داد (همه‌ی سکه‌ها را به رو برگرداند).

۵- ماتریس ۱ و ۱-

تعداد ماتریس‌های $m \times n$ با درایه‌های ۱ و ۱- را پیدا کنید که حاصل‌ضرب عناصر هر سطر آن برابر با ۱- شود. ادعای خود را ثابت کنید.

٦- مرتب کردن دیسک‌ها

تعداد ۲^k دیسک داریم که روی هرکدام، یکی از عددهای ۱ تا ۲^k نوشته شده است. $k + ۲$ میله با شماره‌های صفر تا $k + ۱$ در یک ردیف پست سر هم قرارگرفته‌اند. در ابتدا، دیسک‌ها با یک ترتیب داده‌شده در میله‌ی صفر روی‌هم قرار دارند. در هر حرکت می‌توان بالاترین دیسک موجود روی میله‌ی i ام را برداشته و روی دیسک‌های میله‌ی $i + ۱$ ام قرار داد $(۰ \leq i \leq k)$. این حرکت را با T_i نمایش می‌دهیم. حالت نهایی مرتب، حالتی است که در آن تمام دیسک‌ها به ترتیب از شماره‌ی ۱ تا ۲^k (پایین به بالا) روی میله‌ی شماره‌ی $k + ۱$ قرارگرفته باشند. برای مثال، به ازای $k = ۱$ دنباله‌ی حرکت‌های لازم برای رسیدن به حالت نهایی مرتب از حالت اولیه‌ای به شکل زیر می‌تواند به‌صورت $(T_۰, T_۰, T_۱, T_۱)$ باشد.



ثابت کنید به ازای هر k می‌توان از هر ترتیب اولیه‌ی دیسک‌ها روی میله‌ی شماره‌ی صفر به یک حالت نهایی مرتب رسید.

۷- شبکه‌ی کامپیوتری

در یک شبکه‌ی کامپیوتری، ۲^k کامپیوتر با شماره‌های ۱ تا ۲^k وجود دارد. هر یک از این کامپیوترها با یک کد یکتا که یک دنباله‌ی k تایی از عددهای صفر و یک است، مشخص می‌شود. دو کامپیوتر به‌صورت مستقیم به هم متصل هستند اگر و فقط اگر کد مربوط به آن‌ها دقیقاً در یک رقم با هم تفاوت داشته باشند. برای مثال اگر $k = ۴$ باشد، کامپیوتری که دارای کد ۰۱۰۰ است مستقیماً به کامپیوترهایی با کدهای ۱۱۰۰، ۰۰۰۰، ۰۱۱۰ و ۰۱۰۱ متصل است.

در ابتدای کار، هر یک از کامپیوترها، دارای یک پیام است. پیامی که در ابتدا در کامپیوتر شماره‌ی $i(۱ \leq i \leq ۲^k)$ وجود دارد. باید درنهایت به کامپیوتر $p_i(۱ \leq p_i \leq ۲^k)$ برسد. فرض کنید که در بین p_i ها عدد تکراری وجود ندارد؛ یعنی درنهایت هرکدام از کامپیوترها باید یک پیام دریافت کنند.

در هر مرحله، هرکدام از کامپیوترها می‌تواند پیغامی که دارد را به یکی از کامپیوترهایی که مستقیماً به آن متصل است بدهد؛ به شرطی که هر کامپیوتری پس از پایان آن مرحله بیش از یک پیام نداشته باشد. (اگر در یک مرحله، کامپیوتر a پیام خود را به کامپیوتر b بدهد، کامپیوتر b هم باید پیامی که قبل از این مرحله داشته است را در همین مرحله به یک کامپیوتر بدهد. همچنین هیچ کامپیوتر دیگری غیر از a نمی‌تواند پیام خود را در همین مرحله به b بدهد).

ثابت کنید که در حداکثر $۲^k - ۱$ مرحله، کامپیوترها می‌توانند همه‌ی پیام‌ها را با توجه به‌شرط فوق به مقصدشان برسانند.

۸- سه دستورالعمل

یک کامپیوتر دارای حافظه‌ای است که می‌تواند یک لیست از عددها (که هرکدام از آن‌ها ۰، ۱ یا ۲ هستند) را نگه‌داری کند. محدودیتی در طول لیستی که در حافظه‌ی این کامپیوتر نگه‌داری می‌شود وجود ندارد. این کامپیوتر می‌تواند یک برنامه را اجرا کند. هر برنامه شمال تعدادی دستور است که به ترتیب مشخصی قرارگرفته‌اند. این کامپیوتر تنها سه نوع دستورالعمل را قبول می‌کند که عبارت‌اند از:

- $E\ x$ (x یکی از عددهای ۰، ۱ یا ۲ است.): این دستور، عدد xرا به انتهای (سمت راست) لیست عددها اضافه می‌کند و پس‌ازآن دستور بعدی را انجام می‌دهد.

- D : عددی که در ابتدای (سمت چپ) لیست عددها قرار دارد را از لیست برمی‌دارد .اگر این عدد ۰ بود، دستور بعدی را اجرا می‌کند؛ اگر ۱ بود، یک دستور را جا می‌اندازد و دستور بعدازآن را اجرا می‌کند؛ و اگر ۲ بود، دو دستور را جا می‌اندازد و دستور بعدی را اجرا می‌کند.
- $J\ \ d$ (d یک عدد صحیح مثبت یا منفی است.): اگر d مثبت بود، دستوری که d تا بعد از دستور فعلی است و اگر d منفی بود، دستوری که d تا قبل از دستور فعلی است را اجرا می‌کند.

اجرای برنامه با اجرای دستورالعمل اول آن شروع می‌شود و مطابق با قوانین فوق ادامه می‌یابد. اگر در یک مرحله، دستورالعملی که قرار است اجرا شود، وجود نداشت (برای مثال به یک دستور J به یک دستور که در برنامه وجود ندارد پرش کردیم)، اجرای برنامه متوقف می‌شود. همچنین اگر لیست عددهای خالی بود و به دستورالعمل D برخوردیم، برنامه متوقف می‌شود.

برای مثال برنامه‌ی زیر را در نظر بگیرید. (شماره‌های نوشته‌شده در سمت چپ دستورات، نشان‌دهنده‌ی ترتیب اجرای آن‌هاست.) درصورتی‌که پیش از اجرای این برنامه لیست عددها ۱، ۱ باشد، با اجرای این برنامه به ترتیب دستورات شماره‌ی ۱، ۳، ۵، ٦، ۳ و ٤ اجرا می‌شوند و پس‌ازآن، برنامه متوقف می‌شود. پس از متوقف شدن برنامه، لیست عددها خالی خواهد بود.

۱.D
۲.E ۱
۳.D
۴.J ۴
۵.E ۰
۶.J $-$ ۳

الف) برنامه‌ی زیر را در نظر بگیرید:

۱.E ۲
۲.D
۳.J ۳
۴.J ۴
۵.J ۱۰
۶.E ۱
۷.J $-$ ۵
۸.E ۰
۹.J $-$ ۷

درصورتی‌که قبل از اجرای برنامه، لیست عددها ۰، ۱، ۰، ۰، ۱ باشد، پس از اجرای برنامه این لیست به چه شکلی در خواهد آمد؟ توضیح دهید.

ب) برنامه‌ی زیر را در نظر بگیرید:

۱.E	۲	۱۰.E	۰	۱۹.J	−۱۵
۲.E	۲	۱۱.J	−۸	۲۰.J	−۱۵
۳.D		۱۲.E	۲	۲۱.J	۱۰
۴.J	۵	۱۳.E	۱	۲۲.E	۰
۵.J	۱	۱۴.D		۲۳.J	−۹
۶.E	۲	۱۵.J	۷	۲۴.E	۱
۷.E	۰	۱۶.J	۸	۲۵.J	−۱۱
۸.J	۶	۱۷.E	۲		
۹.D		۱۸.D			

درصورتی‌که قبل از اجرای این برنامه، لیست عددها از ۱۳۷۶ تا عدد صفر تشکیل‌شده باشد، پس از اجرای این برنامه این لیست به چه شکلی در خواهد آمد؟ توضیح دهید.

ج) فرض کنید که یک لیست از عددهای ۰ و ۱ در حافظه‌ی این کامپیوتر قرار دارد. (توجه کنید که لیست، شامل عدد ۲ نیست.) برنامه‌ای برای این کامپیوتر بنویسید که پس از اجرای آن، این لیست برعکس شود. در مورد برنامه‌ای که می‌نویسید توضیح دهید.