فصل ۱

الگوریتم چیست؟

عنوان «الگوریتم» به شیوه و روشی اطلاق می‌شود که برای محاسبات کامپیوتری و یا پردازش اطلاعات با استفاده از کامپیوتر به‌کار گرفته می‌شود. هر الگوریتم دنباله‌ای از مقادیر را به عنوان ورودی می‌گیرد، با شیوه‌ای مشخص آنها را پردازش می‌کند و نتیجه را به عنوان خروجی بیرون می‌دهد.

واژه‌ی الگوریتم از نام محمدبن موسی خوارزمی گرفته شده است که ریاضیدانی ایرانی بود و در قرن سوم هجری می‌زیست. او به الخوارزمی مشهور بود و واژه‌ی Algorithm به معنای شیوه و روش محاسبه از این عنوان گرفته شده است. شاخه‌ای از ریاضیات که جبر نامیده می‌شود اولین بار در کتاب «الجبر و المقابله»ی او مطرح شده است. خوارزمی در این کتاب برای حل مسائل کاربردی شیوه‌هایی با استفاده از معادلات ارائه کرده است و همین امر سبب شده است که شیوه‌های محاسبه و پردازش اطلاعات هم به «الگوریتم» مشهور شوند.

یکی از مسائل کتاب جبر و مقابله‌ی خوارزمی و راه‌حل آن به قرار زیر است:

اگر بگویی ده را به دو قسمت تقسیم کردم، پس از آن هر قسمت را در خودش ضرب نمودم، و سپس حاصلضرب هر دو را جمع کردم، پنجاه و هشت درهم شد. راه‌حل آن چنین است: یکی از قسمت‌ها را شیء فرض می‌کنی و دیگری را ده منهای شیء.

صورت مسئله به زبان ریاضیات امروز چنین می‌شود:

$$x = \text{شیء}$$

$$x^۲ + (۱۰ - x)^۲ = ۵۸$$

پس از آن صد به اضافه‌ی دو مال را با بیست شیء ناقص جبر می‌کنی، و پنجاه و هشت را بر آن می‌افزایی

$$x^۲ = \text{مال}$$
$$۱۰۰ + ۲x^۲ = \text{صد به اضافه‌ی دو مال}$$
$$-۲۰x = \text{بیست شیء ناقص}$$
$$۲x^۲ + ۱۰۰ = ۵۸ + ۲۰x$$

آنگاه این دو مال را به مال واحد تبدیل می‌کنی

$$x^۲ + ۵۰ = ۲۹ + ۱۰x$$

سپس آن را مقابله می‌کنی

$$x^۲ + ۲۱ = ۱۰x$$

روی جلد کتاب جبر و مقابله‌ی خوارزمی

پس از آن جذر (= ضریب x) را نصف می‌کنی می‌شود پنج؛ این عدد را در خودش ضرب می‌کنی می‌شود بیست و پنج، از این عدد بیست و یک را که همراه مال بود کم می‌کنی چهار باقی می‌ماند، جذر آن را می‌گیری دو می‌شود، این عدد را از نصف جذرها که مقدارش پنج است، کم می‌کنی سه باقی می‌ماند، عدد سه یکی از این دو قسمت ده است، قسمت دیگرش هفت است.

۱-۱ حل معادله‌ی درجه‌ی دوم

روش حل معادلات درجه‌ی دوم از اولین روش‌های الگوریتمی است که با آنها آشنا می‌شویم. ریشه‌های معادله‌ی درجه‌ی دومی به صورت $ax^۲+bx+c=۰$ (که $a \neq ۰$ و $b^۲ - ۴ac \geq ۰$) از دستورهای زیر به دست می‌آیند:

$$x_۱ = \frac{-b + \sqrt{b^۲ - ۴ac}}{۲a} \quad \text{و} \quad x_۲ = \frac{-b - \sqrt{b^۲ - ۴ac}}{۲a}$$

اگر a، b، و c مشخص باشند، می‌توانیم $x_۱$ و $x_۲$ را به ترتیب زیر محاسبه کنیم.

- حل معادله‌ی درجه‌ی دوم

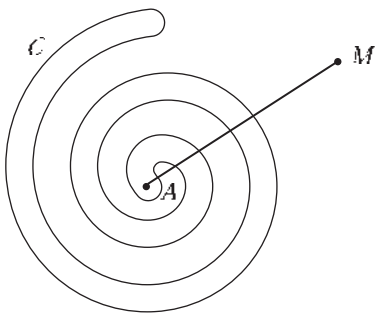
۱. a، b، و c را از ورودی بگیر.
۲. اگر $a = ۰$،
۳. پیغام بده «خطا: ضریب درجه‌ی دو صفر است».

۴. برو به ۱۳
۵. $\Delta = b^2 - 4ac$ را محاسبه کن.
۶. اگر $\Delta < 0$،
۷. پیغام بده «معادله ریشه‌ی حقیقی ندارد».
۸. برو به ۱۳
۹. جذر Δ، $\sqrt{\Delta}$ را محاسبه کن.
۱۰. $\sqrt{\Delta}$ را با $-b$ جمع کن و نتیجه را بر $2a$ تقسیم کن تا x_1 به‌دست بیاید.
۱۱. $\sqrt{\Delta}$ را از $-b$ کم کن و نتیجه را بر $2a$ تقسیم کن تا x_2 به‌دست بیاید.
۱۲. x_1 و x_2 را در خروجی قرار بده.
۱۳. پایان.

مثال فوق نمونه‌ای از یک الگوریتم است: a و b و c ورودی هستند، با آنها محاسبه و پردازش صورت می‌گیرد، و x_1 و x_2 در خروجی داده می‌شوند. مسئله‌های گوناگون دیگری هم وجود دارند که با الگوریتم‌های مناسب حل می‌شوند. برای طراحی الگوریتم‌ها روش‌های مختلفی وجود دارد که در این کتاب با بعضی از آنها آشنا می‌شویم؛ امّا قبل از اینکه به این تکنیک‌ها بپردازیم با بررسی چند مثال با تفکر الگوریتمی بیشتر آشنا می‌شویم.

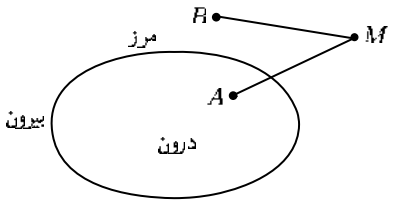
۱-۲ داخل و خارج ناحیه‌ی مارپیچی

خم بسته‌ی C در صفحه به شکل زیر داده شده است.



می‌خواهیم ببینیم نقطه‌ی دلخواهی در صفحه در ناحیه‌ی درونی این خم قرار دارد و یا در ناحیه‌ی بیرونی آن. برای طرح الگوریتمی برای حل این مسئله از این ویژگی استفاده می‌کنیم که هر خم بسته

صفحه را به دو ناحیه‌ی درون و بیرون تقسیم می‌کند که با مرز از هم جدا شده‌اند.



برای اینکه ببینیم نقطه‌ی داده شده‌ای در ناحیه‌ی درونی خم C قرار دارد یا نه، نقطه‌ای مانند M را در ناحیه‌ی بیرونی در نظر می‌گیریم و آن را به نقطه‌ی A وصل می‌کنیم. اگر تعداد دفعات تقاطع MA با مرز عددی فرد باشد، A در درون و در غیر این صورت در ناحیه‌ی بیرونی قرار دارد. پس به الگوریتم زیر می‌رسیم:

• ناحیه‌ی داخل و خارج

۱. خم بسته‌ی C و نقطه‌ی A را از ورودی بگیر.
۲. اگر A روی C است،
۳. پیغام بده «نقطه روی خم است»
۴. در غیراین صورت،
۵. نقطه‌ی M را بیرون C در نظر بگیر.
۶. تعداد نقاط برخورد پاره‌خط MA با C را محاسبه کن و نتیجه را در n بگذار.
۷. به تعداد نقاطی که MA در آنها بر C مماس است به n اضافه کن.
۸. اگر n فرد است،
۹. پیغام بده «نقطه درون خم است».
۱۰. در غیراین صورت،
۱۱. پیغام بده «نقطه بیرون خم است».
۱۲. پایان.

۱-۳ خُرد کردن پول

می‌خواهیم n تومان را با سکه‌های ۱، ۲، و ۵ تومانی طوری خرد کنیم که تعداد سکه‌ها کمترین تعداد ممکن باشد. البته فرض می‌کنیم n عددی طبیعی است و از این سکه‌ها هر تعداد که لازم باشد، در اختیار داریم. الگوریتم مورد نظر را در صفحه‌ی بعد آورده‌ایم.

• خرد کردن پول

۱. n را از ورودی بگیر.

۲. n را به ۵ تقسیم کن، خارج قسمت را در n۵ و باقیمانده را در r۵ قرار بده.

۳. r۵ را به ۲ تقسیم کن، خارج قسمت را در n۲ و باقیمانده را در n۱ قرار بده.

۴. پیغام بده «n۵ سکه‌ی پنج تومانی، n۲ سکه‌ی دو تومانی و n۱ سکه‌ی ۱ تومانی»

۵. پایان.

۱-۴ بزرگ‌ترین مقسوم‌علیه مشترک

بزرگ‌ترین مقسوم‌علیه مشترک دو عدد صحیح و مثبت، بزرگ‌ترین عددی است که هر دوی آنها بر آن قابل قسمت‌اند. الگوریتم اقلیدس براساس قضیه‌ی زیر شکل گرفته است:

قضیه. اگر a و b دو عدد صحیح و مثبت باشند که $a \geq b$، آنگاه (a,b)ب‌م‌م $=$ (r,b)ب‌م‌م که r باقیمانده‌ی تقسیم a بر b است.

در واقع روش معروف نردبانی که همان الگوریتم اقلیدس است براساس این قضیه شکل گرفته است. مثلاً برای پیدا کردن ب‌م‌م دو عدد ۲۴ و ۱۵ به شیوه‌ی زیر عمل می‌کنیم.

	۱	۱	۱	۲
۲۴	۱۵	۹	۶	۳
۹	۶	۳	۰	

کار را آن‌قدر ادامه می‌دهیم تا به باقیمانده‌ی صفر برسیم و آخرین عدد، یعنی ۳، ب‌م‌م ۲۴ و ۱۵ است.
الگوریتم اقلیدس در حالت کلی به صورت زیر بیان می‌شود:

• ب‌م‌م

۱. a و b را از ورودی بگیر.

۲. اگر $a < b$،

۳. مقدار a و b را عوض کن.

۴. a را بر b تقسیم کن و باقیمانده را در r قرار بده.

۵. اگر $r \neq ۰$،
۶. $a \leftarrow b$
۷. $b \leftarrow r$
۸. برو به ۲
۹. b را در خروجی قرار بده.
۱۰. پایان.

الگوریتم فوق را برای دو عدد ۲۴ و ۱۵ اجرا می‌کنیم:

مرحله	۱	۲	۳	۴
a	۲۴	۱۵	۹	۶
b	۱۵	۹	۶	③
r	۹	۶	۳	۰

همان‌طور که می‌بینید، با تکرار عمل تقسیم و باقیمانده‌گیری آن‌قدر جلو می‌رویم تا باقیمانده برابر صفر شود. آخرین مقدار که به b نسبت داده‌ایم برابر ب‌م‌م است.

نکته‌ی دیگری که در الگوریتم فوق قابل توجه است این است که از علامت $\leftarrow$ برای نسبت دادن استفاده کرده‌ایم. وقتی می‌نویسیم $a \leftarrow b$ یعنی آخرین مقداری که به b نسبت داده شده بود به a نسبت داده شود و در نتیجه مقدار قبلی نسبت داده شده به a از بین می‌رود. به عبارت دیگر، در اینجا سه خانه از حافظه را با اسامی a، b، و r نام‌گذاری کرده‌ایم که در مرحله‌ی شروع مثلاً مقادیر ۲۴ و ۱۵ به‌ترتیب در خانه‌های a و b قرار می‌گیرند.

$$a : \boxed{۲۴} \qquad b : \boxed{۱۵}$$

سپس a بر b تقسیم می‌شود و باقیمانده‌ی آن در خانه‌ی r وارد می‌شود.

$$r : \boxed{۹}$$

حال مقداری را که در خانه‌ی b بود به خانه‌ی a می‌بریم و مقداری را که در خانه‌ی r بود به خانه‌ی b نسبت می‌دهیم.

$$a : \boxed{۱۵} \qquad b : \boxed{۹}$$

و کار به همین ترتیب ادامه پیدا می‌کند.

کارگاه

می‌گویند شخصی که شطرنج را از هندوستان به عنوان تحفه به دربار خسرو انوشیروان آورده بود وقتی از او خواستند به پاس این کار پاداشی درخواست کند، چنین خواسته‌ای را مطرح کرد که به او مقداری گندم داده شود به این ترتیب که در خانه‌ی اول شطرنج یک دانه‌ی گندم، در خانه‌ی دوم دو دانه‌ی گندم، در خانه‌ی سوم چهار دانه‌ی گندم، و به همین ترتیب در هر خانه‌ی بعدی تعداد دانه‌های گندم دوبرابر شود، تا خانه‌ی ۶۴ام، و مجموع این مقادیر گندم به او داده شود. الگوریتمی طراحی کنید تا با روش تکرار، مجموع تعداد دانه‌های گندم را محاسبه کند.

۱-۵ مرتب‌سازی سه عدد

فرض کنید سه عدد x، y و z را از ورودی گرفته‌ایم و می‌خواهیم این اعداد را به‌ترتیب از کوچک به بزرگ در خروجی بنویسیم. به الگوریتم زیر توجه کنید.

- مرتب‌سازی سه عدد

۱. x، y و z را از ورودی بگیر.
۲. اگر $x > y$،
۳. اگر $y > z$،
۴. در خروجی قرار بده z، y ، x.
۵. در غیراین صورت،
۶. اگر $x > z$،
۷. در خروجی قرار بده y، z، x.
۸. در غیراین صورت،
۹. در خروجی قرار بده y، x، z.
۱۰. در غیراین صورت،
۱۱. اگر $x > z$،
۱۲. در خروجی قرار بده z، x، y.

۱۳. در غیراین صورت،
۱۴. اگر $y > z$،
۱۵. در خروجی قرار بده x، z، y.
۱۶. در غیراین صورت،
۱۷. در خروجی قرار بده x، y، z.
۱۸. پایان.

در این مثال برای مرتب‌سازی سه عدد در واقع همه‌ی حالت‌های ممکن را بررسی کردیم، ولی اگر تعداد اعدادی که می‌خواهیم مرتب کنیم از سه عدد بیشتر باشند در نظر گرفتن همه‌ی حالت‌ها بسیار وقت‌گیر و دشوار است. در این الگوریتم، تنها تعداد دستورهای خروجی برابر است با تعداد جایگشت‌های اعضای دنباله‌ی ورودی که در این مثال برابر است با $۶ = !۳$. با کمی دقت، می‌توانیم تعداد مراحل الگوریتم را کمتر کنیم:

• مرتب‌سازی سه عدد، نسخه‌ی ۲

۱. x و y و z را از ورودی بگیر.
۲. اگر $y < x$،
۳. مقدار x و y را عوض کن.
۴. اگر $z < y$،
۵. مقدار y و z را عوض کن.
۶. اگر $y < x$،
۷. مقدار x و y را عوض کن.
۸. در خروجی قرار بده x، y، z.
۹. پایان.

ایده‌ی این الگوریتم، «هُل دادن» مقدارهای بزرگ‌تر به سمت متغیرهای «انتهایی» است و به همین خاطر، به مرتب‌سازی حبابی مشهور شده است.

پیش از ارائه‌ی صورت کلی الگوریتم مرتب‌سازی حبابی، لازم است با ساختار آرایه‌ها آشنا شویم.

آرایه‌ها

آرایه ساختاری است برای نگهداری دنباله‌ای متناهی از مقادیر یا داده‌ها. آرایه، مانند دنباله، از تعدادی «خانه»ی پشت سر هم تشکیل شده است و مقادیر یا داده‌ها درون این خانه‌ها قرار می‌گیرند. مقدار هریک از این خانه‌ها با یک اندیس مشخص می‌شود که شماره‌ی آن خانه است. به‌عنوان مثال، آرایه‌ی شش‌خانه‌ای A به صورت زیر است:

	۱	۲	۳	۴	۵	۶
A :						

عددی که در بالای هر خانه نوشته شده است، اندیس یا نشانه‌ی آن خانه است. اگر A را مثلاً به صورت

	۱	۲	۳	۴	۵	۶
A :	۷	۳	۵	۲	۱۱	۸

پر کنیم، می‌نویسیم $A[۱] = ۷$، $A[۲] = ۳$ و به همین ترتیب تا $A[۶] = ۸$.

این آرایه را به صورت

$$A : \langle ۷, ۳, ۵, ۲, ۱۱, ۱۸ \rangle$$

هم می‌توانیم نشان بدهیم.

مقصود از زیرآرایه نیز تعدادی از خانه‌های متوالی آرایه‌ی بزرگ‌تر است، مثلاً

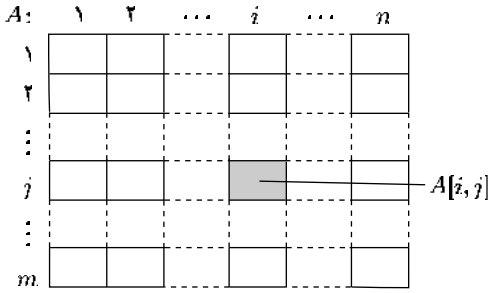
۷	۳	۵

زیرآرایه‌ای از آرایه‌ی A است.

می‌توانیم آرایه‌ی دوبعدی را نیز به‌عنوان ساختاری ویژه برای ساماندهی و دسترسی به مقادیر یا داده‌ها استفاده کنیم. آرایه‌ی دوبعدی نظیر جدولی است که در خانه‌های آن مقادیر مورد نظر قرار دارد و هر خانه را با دو اندیس سطر و ستون مشخص می‌کنیم. به‌عنوان مثال، آرایه‌ی دوبعدی A که به صورت

A :	۱	۲	۳	۴
۱	۱۲۰	۸۳	۱۹	۹۱
۲	۲۷	۳۲	۴۸	۳۱
۳	۱۴	۱۱	۹۷	۲۹

نشان داده شده، آرایه‌ای ۳ در ۴ است. در اینجا مثلاً $A[۱,۱] = ۱۲۰$، $A[۱,۲] = ۸۳$ و یا $A[۲,۳] = ۴۸$ و بقیه‌ی خانه‌ها نیز به همین ترتیب مشخص می‌شوند. در حالت کلی، آرایه‌ی دوبعدی m در n از یک جدول مقادیر با m سطر و n ستون تشکیل می‌شود.



هر خانه‌ی آرایه با $A[i,j]$ نشان داده می‌شود که مقصود خانه‌ی واقع در سطر iام و ستون jام است.

۱-۶ مرتب‌سازی حُبابی

اصولاً، مقصود از مرتب‌سازی دنباله‌ای از اعداد این است که آنها را از کوچک به بزرگ (یا برعکس) مرتب کنیم. مثلاً دنباله‌ی

$$۷, ۳, ۵, ۲, ۱۱, ۸$$

پس از مرتب شدن به‌صورت

$$۲, ۳, ۵, ۷, ۸, ۱۱$$

درمی‌آید. در حالت کلی n عدد مانند $a_۱$، $a_۲$، ...، a_n را در نظر می‌گیریم. مقصود از مرتب‌سازی آنها پیدا کردن جایگشتی مانند σ روی $\{۱, \ldots, n\}$ است که اعداد در این جایگشت مثلاً به شکل غیرنزولی مرتب شده باشند، یعنی

$$a_{\sigma(۱)} \leq a_{\sigma(۲)} \leq \cdots \leq a_{\sigma(n)}$$

در الگوریتم مرتب‌سازی حبابی هر عضو با عضو بعدی مقایسه می‌شود و در هر مقایسه عضو بزرگ‌تر را جابه‌جا می‌کنیم و این کار را ادامه می‌دهیم تا دنباله مرتب شود. به مثالی توجه کنید:

⑦	۳	۵	۲	۱۱	۸
۳	⑦	۵	۲	۱۱	۸
۳	۵	⑦	۲	۱۱	۸
۳	۵	۲	⑦	۱۱	۸

۳	۵	۲	۷	(۱۱)	۸
(۳)	۵	۲	۷	۸	۱۱
۳	(۵)	۲	۷	۸	۱۱
۳	۲	۵	۷	۸	۱۱
۳	۲	۵	۷	(۸)	۱۱
(۳)	۲	۵	۷	۸	۱۱
۲	۳	۵	۷	۸	۱۱
۲	(۳)	۵	۷	۸	۱۱
۲	۳	(۵)	۷	۸	۱۱
۲	۳	۵	(۷)	۸	۱۱
۲	۳	۵	۷	(۸)	۱۱
۲	۳	۵	۷	۸	۱۱

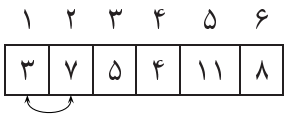
• مرتب‌سازی حبابی

۱. مقادیر n و $A[۱]$، $A[۲]$، ...، $A[n]$ را از ورودی بگیر.
۲. به‌ازای $i = n - ۱$ تا ۱:
۳. به‌ازای $j = ۱$ تا i:
۴. اگر $A[j] > A[j + ۱]$،
۵. مقدار $A[j]$ و $A[j + ۱]$ را عوض کن.
۶. پایان.

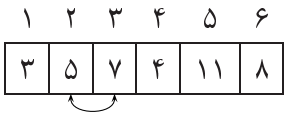
این الگوریتم را برای مثالی که قبلاً آوردیم بررسی می‌کنیم. در اینجا $n = ۶$ و در دور اول $i = ۵$، و j از ۱ تا ۵ تغییر می‌کند.

	۱	۲	۳	۴	۵	۶
A	۷	۳	۵	۴	۱۱	۸

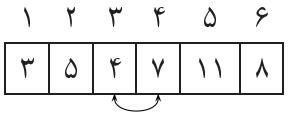
به‌ازای $j = ۱$ مقادیر $A[۱]$ و $A[۲]$ مقایسه می‌شوند و چون $A[۱]$ بزرگ‌تر است جای آنها عوض می‌شود:



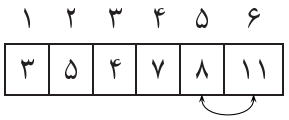
سپس $A[۲]$ با $A[۳]$ مقایسه می‌شود و بازهم جای آنها عوض می‌شود:



در ادامه $A[۳]$ و $A[۴]$ را مقایسه می‌کنیم:



در مقایسه‌ی بعدی، چون $A[۵]$ از $A[۴]$ بزرگ‌تر است، جابه‌جایی نداریم؛ ولی در مقایسه‌ی $A[۵]$ و $A[۶]$ مجدداً آنها را جابه جا می‌کنیم:



تا اینجا، یک مرحله از کار انجام شده است و بزرگ‌ترین عدد به خانه‌ی آخر یعنی $A[۶]$ رفته است. این کار را طبق الگوریتم ادامه می‌دهیم؛ یعنی در مرحله‌ی بعد $i = ۴$ و j از ۱ تا ۴ تغییر می‌کند تا بزرگ‌ترین عضو بقیه‌ی دنباله در انتهای همان زیردنباله قرار بگیرد، و کار به همین ترتیب ادامه پیدا می‌کند تا تمام شود.

مرتب‌سازی از مسئله‌های مهم طراحی الگوریتم‌ها است و روش‌های مختلفی برای مرتب‌سازی وجود دارد که در آینده با آنها آشنا خواهیم شد. تعداد مقایسه‌های مورد نیاز در هر روش، معیاری برای ارزیابی آنها محسوب می‌شود. در مرتب‌سازی حبابی چند مقایسه انجام می‌دهیم؟ در دور اول $n - ۱$ مقایسه، در دور دوم $n - ۲$ مقایسه، تا بالاخره در دور آخر ۱ مقایسه صورت می‌گیرد؛ پس تعداد مقایسه‌های لازم برابر است با

$$(n-۱) + (n-۲) + \cdots + ۱ = \frac{۱}{۲}(n-۱)n = \frac{۱}{۲}n^۲ - \frac{۱}{۲}n$$

در واقع تعداد مقایسه‌ها از مرتبه‌ی $n^۲$ است.

کارگاه

الگوریتمی بنویسید که از بین n عدد $A[۱]$، $A[۲]$، ...، $A[n]$ بزرگ‌ترین و کوچک‌ترین آنها را پیدا کند (بدون استفاده از مرتب‌سازی).

تمرین

۱. چند ردیف اول مثلث خیام-پاسکال به‌صورت زیر است:

۱
۱ ۱
۱ ۲ ۱
۱ ۳ ۳ ۱
۱ ۴ ۶ ۴ ۱

الگوریتمی بنویسید که عدد n را از ورودی بگیرد و ضرایب موجود در سطر nام مثلث خیام-پاسکال را محاسبه کند و در خروجی بنویسد.

۲. الگوریتمی برای حل دستگاه دو معادله و دو مجهول زیر بنویسید:

$$ax + by = e$$
$$cx + dy = f$$

مقادیر a، b، c، d، e و f از ورودی گرفته می‌شوند و x و y باید در خروجی قرار داده شوند.

۳. اعداد ۱ تا ۱۰ در مبنای ۲ به‌صورت زیر هستند:

۰, ۱, ۱۰, ۱۱, ۱۰۰, ۱۰۱, ۱۱۰, ۱۱۱, ۱۰۰۰, ۱۰۰۱, ۱۰۱۰

عدد ۳۲۱ در مبنای ۱۰ در واقع عبارت است از

$$۳۲۱ = ۳ \times ۱۰^{۲} + ۲ \times ۱۰ + ۱$$

عدد ۱۰ نیز در مبنای ۲ عبارت است از

$$۱۰ = (۱۰۱۰)_{۲} = ۱ \times ۲^{۳} + ۰ \times ۲^{۲} + ۱ \times ۲^{۱} + ۰$$

الف) الگوریتمی برای تبدیل یک عدد از مبنای ۲ به مبنای ۱۰ بنویسید.

ب) الگوریتمی برای تبدیل یک عدد از مبنای ۱۰ به مبنای ۲ بنویسید.

ج) الگوریتمی برای جمع دو عدد در مبنای ۲ بنویسید.

۴. خط $d: y = k$ و نقطه‌ی $M(x, y)$ مفروض‌اند. الگوریتمی بنویسید که تعیین کند نقطه‌ی M بالا، پایین و یا روی خط d قرار دارد.

۵. مجموعه‌ی $\{۱, ۲, \ldots, ۱۰\}$ مفروض است، الگوریتمی بنویسید تا تعداد زیرمجموعه‌هایی از این مجموعه را که مجموع عناصر آنها عددی زوج است تعیین کند.

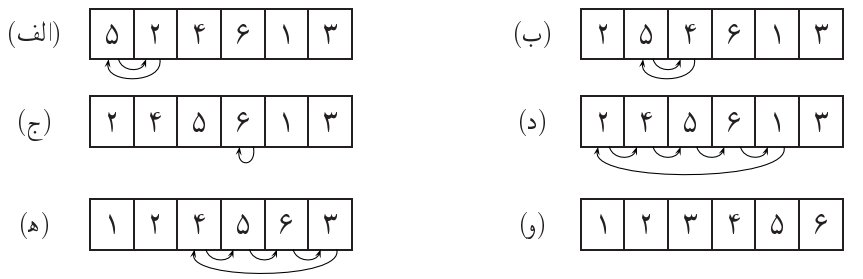
۶. الگوریتم مرتب‌سازی حبابی را به طور برعکس بنویسید، یعنی آن را طوری بنویسید که بین اعداد $a_۱$، $a_۲$، ...، a_n از آخر شروع کند و دنباله را مرتب کند.

۷. الگوریتمی بنویسید که همه‌ی زیرمجموعه‌های مجموعه‌ی $\{۱, ۲, \ldots, ۱۰\}$ را پیدا کند و در خروجی بنویسد.

۸. الگوریتمی بنویسید که همه‌ی جایگشت‌های $\{۱, ۲, ۳, ۴, ۵\}$ را پیدا کند و در خروجی بنویسد.

۹. الگوریتمی بنویسید که وضعیت یک خط و یک دایره را مشخص کند، یعنی مشخص کند که خط دایره را قطع نمی‌کند، بر آن مماس است، و یا آن را در دو نقطه قطع می‌کند.

۱۰. یکی دیگر از روش‌های مرتب‌سازی مجموعه‌ای از اعداد، مرتب‌سازی درجی است. در این روش مقادیر را در یک آرایه قرار می‌دهیم و از خانه‌ی دوم آرایه شروع می‌کنیم و در هر مرحله، مقدار موجود در هر خانه را با مقادیر سمت چپ خودش مقایسه می‌کنیم و با جابه جا کردن مقادیر، آنها را در جای مناسب قرار می‌دهیم تا در پایان کار آرایه مرتب شود؛ مثلاً برای آرایه‌ی $A = \langle ۵, ۲, ۴, ۶, ۱, ۳ \rangle$،



الف) الگوریتمی برای مرتب‌سازی درجی بنویسید.

ب) آیا می‌توانید تعداد مقایسه‌های لازم برای مرتب‌سازی درجی را تخمین بزنید؟