

به نام خدا

مبانی برنامه نویسی

آرش شفیعی



الگوریتم

- کلمهٔ الگوریتم از نام دانشمند ایرانی محمدبن موسی الخوارزمی گرفته شده است.
- خوارزم منطقه‌ای است در آسیای مرکزی که در حال حاضر در ازبکستان و ترکمنستان قرار دارد و در کنار دریاچهٔ آرال (دریاچهٔ خوارزم) قرار گرفته است. خوارزمی کتاب الجبروالمقابله را نیز به تألیف رسانده است که کلمه جبر¹ در زبان انگلیسی نیز از همین کتاب گرفته شده است.
- تا سال ۱۹۵۰ کلمهٔ الگوریتم بیشتر برای الگوریتم اقلیدس² برای پیدا کردن بزرگ‌ترین مقسوم‌علیه مشترک³ دو عدد به‌کار می‌رفت که در کتاب اصول اقلیدس⁴ توصیف شده است.

¹ algebra

² Euclid's algorithm

³ greatest common divisor

⁴ Euclid's Element

- الگوریتم پیدا کردن بزرگ‌ترین مقسوم‌علیه مشترک را می‌توانیم به صورت زیر وصف کنیم.
۱. (پیدا کردن باقیمانده.) عدد m را بر n تقسیم می‌کنیم. فرض کنید باقیمانده r باشد خواهیم داشت $0 \leq r < n$
 ۲. (آیا باقیمانده صفر است؟) اگر $r = 0$ ، الگوریتم پایان می‌یابد و n جواب مسئله است.
 ۳. (کاهش.) قرار می‌دهیم $m \leftarrow n$ و $n \leftarrow r$ و به مرحله ۱ می‌رویم.

- الگوریتم در واقع یک روند¹ یا دستورالعمل² برای حل یک مسئله محاسباتی است.
- به طور غیر رسمی می‌توانیم بگوییم یک الگوریتم در واقع یک روند محاسباتی گام‌به‌گام است که مجموعه‌ای از مقادیر را که ورودی الگوریتم نامیده می‌شوند دریافت می‌کند و مجموعه‌ای از مقادیر را که خروجی الگوریتم نامیده می‌شوند در زمان محدود تولید می‌کند. بنابراین یک الگوریتم دنباله‌ای است از گام‌های محاسباتی که ورودی‌ها را به خروجی تبدیل می‌کند.

¹ procedure

² recipe

- می‌توان گفت یک الگوریتم ابزاری است برای حل یک مسئله محاسباتی معین.
- یک مسئله با تعدادی گزاره رابطه بین ورودی‌ها و خروجی‌ها را در حالت کلی مشخص می‌کند. یک نمونه از مسئله، در واقع با جایگذاری اعداد و مقادیر برای مسئله کلی به دست می‌آید. یک الگوریتم روشی گام‌به‌گام را شرح می‌دهد که با استفاده از آن در حالت کلی برای همه نمونه‌های یک مسئله، خروجی‌ها با دریافت ورودی‌ها تولید شوند. بنابراین روند یک الگوریتم در رابطه بین ورودی‌ها و خروجی‌ها صدق می‌کند.
- به عنوان مثال، فرض کنید می‌خواهید دنباله‌ای از اعداد را با ترتیب صعودی مرتب کنید. این مسئله که مسئله مرتب سازی¹ نام دارد، یک مسئله بنیادین در علوم کامپیوتر به حساب می‌آید که منشأ به وجود آمدن بسیاری از روش‌های طراحی الگوریتم نیز می‌باشد.

¹ sorting problem

- مسئله مرتب سازی را به طور رسمی به صورت زیر تعریف می‌کنیم.
- ورودی مسئله مرتب سازی عبارت است از دنباله‌ای از n عدد به صورت $\langle a_1, a_2, \dots, a_n \rangle$ و خروجی مسئله عبارت است از دنباله‌ای به صورت $\langle a'_1, a'_2, \dots, a'_n \rangle$ که از جابجا کردن عناصر دنباله ورودی به دست آمده است به طوری که $a'_1 \leq a'_2 \leq \dots \leq a'_n$.
- بنابراین به ازای دنباله ورودی $\langle 58, 42, 36, 42 \rangle$ دنباله خروجی $\langle 36, 42, 42, 58 \rangle$ جواب مسئله است.
- یک نمونه از یک مسئله¹ تشکیل شده است از یک ورودی معین و شرح ویژگی خروجی مسئله. بنابراین دنباله ورودی $\langle 58, 42, 36, 42 \rangle$ به علاوه شرح مسئله مرتب سازی یک نمونه از مسئله مرتب سازی نامیده می‌شود.

¹ instance of a problem

- بنابراین به طور خلاصه، یک مسئله تشکیل شده است از (۱) شرحی از چندین پارامتر یا متغیر آزاد، و (۲) شرحی از ویژگی‌هایی که جواب مسئله دارد.
- یک پارامتر یا متغیر آزاد کمیتی است که مقدار آن مشخص نشده و توسط حروف و یا کلمات، نامی بر آن نهاده شده است.
- یک نمونه مسئله با تعیین مقادیر پارامترهای مسئله به دست می‌آید.
- یک الگوریتم، روندی گام به گام است برای پیدا کردن جواب یک مسئله است.

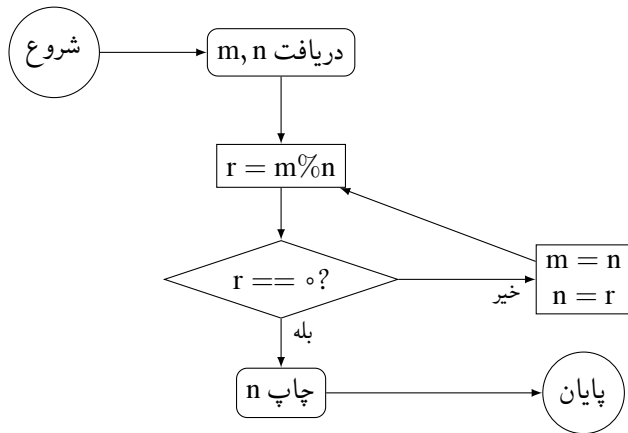
- سرعت اجرای مسئله مرتب سازی به اندازه ورودی یعنی تعداد عناصر دنباله نامرتب و روند الگوریتم بستگی دارد.
- الگوریتم‌های زیادی برای حل مسئله مرتب سازی وجود دارند که هر کدام می‌توانند مزایا و معایبی داشته باشند. به طور مثال یک الگوریتم از میزان حافظه بیشتری استفاده می‌کند، اما زمان کمتری برای محاسبه نیاز دارد و الگوریتم دیگر با میزان حافظه کمتر در زمان بیشتری محاسبه می‌شود که به فراخور نیاز می‌توان از یک از الگوریتم‌ها استفاده کرد.
- عوامل دیگری مانند معماری کامپیوتر، نوع پردازنده و میزان حافظه نیز در زمان اجرای یک الگوریتم مؤثرترند اما این عوامل فیزیکی هستند و صرف نظر از عوامل فیزیکی می‌توان الگوریتم‌ها را از لحاظ میزان حافظه مورد نیاز و زمان اجرا با یکدیگر مقایسه کرد.

- یک الگوریتم برای یک مسئله محاسباتی درست است اگر به ازای هر نمونه از مسئله که با تعدادی ورودی معین شده است، (۱) توقف کند، بدین که در زمان محدود به اتمام برسد و (۲) خروجی تعیین شده توسط شرح مسئله را تولید کند. یک الگوریتم درست در واقع یک مسئله محاسباتی را حل می‌کند.
- یک الگوریتم نادرست ممکن است به ازای برخی از ورودی‌ها توقف نکند یا ممکن است به ازای برخی از ورودی‌ها خروجی نادرست تولید کند.
- یک الگوریتم را می‌توان با استفاده از یک زبان طبیعی مانند فارسی یا انگلیسی توصیف کرد و یا برای توصیف آن از یک برنامه کامپیوتری یا یک زبان ساده شده مانند فلوچارت یا شبه‌کد استفاده کرد. تنها نیازمندی یک الگوریتم توصیف دقیق گام‌های الگوریتم است و زبان مورد استفاده برای توصیف اهمیتی ندارد.

- یک الگوریتم را به صورت یک فلوچارت¹ می‌توانیم رسم کنیم.
- یک فلوچارت یا روندنما نموداری است که روند انجام کاری را نشان می‌دهد.
- یک فلوچارت، الگوریتم را به صورت تصویری به نمایش می‌گذارد. در یک فلوچارت معمولاً برای گام‌های محاسباتی از مستطیل و برای گام‌های شرطی یا بیضی یا لوزی استفاده می‌شود. همچنین در گام‌هایی که ورودی از کاربر گرفته می‌شود یا خروجی برای نمایش به کاربر چاپ می‌شود از شکل‌هایی مانند متوازی‌الاضلاع یا شکل‌های قراردادی دیگر شبیه به مستطیل استفاده می‌شود. هر گام به گام بعدی توسط یک علامت فلش متصل می‌شود. شروع و پایان الگوریتم را معمولاً با دایره نشان می‌دهند.

¹ flowchart

- برای مثال الگوریتم اقلیدس را می توان به صورت زیر رسم کرد.



- در الگوریتم‌ها معمولاً از علامت \leftarrow یا $=$: یا $=$ برای عملیات انتساب استفاده می‌شود. برای مثال $m \leftarrow n$ یا $m = n$ یعنی m را با مقدار فعلی n مقدار دهی می‌کنیم.
- معمولاً از علامت $==$ برای تساوی استفاده می‌شود. برای مثال می‌توانیم بپرسیم آیا مقدار m برابر است با مقدار n و می‌نویسیم اگر $m == n$ یا $m == n?$.
- به عنوان مثال دیگر، برای افزایش مقدار یک متغیر به اندازه یک واحد می‌نویسیم $n \leftarrow n + 1$ یعنی مقدار n برابر است با مقدار فعلی n به علاوه یک. معمولاً این عبارت را به این صورت می‌خوانیم: مقدار n برابر می‌شود با $n+1$. همچنین می‌توانیم بنویسیم $n = n + 1$ که این عملیات انتساب با عملگر تساوی در ریاضی متفاوت است.

- در نشانه گذاری ریاضی معمولاً دنباله‌ها را با استفاده از اندیس‌ها نمایش می‌دهیم برای مثال دنباله v_1, v_2, \dots, v_n یک دنباله از n متغیر است. در الگوریتم‌ها معمولاً از عملگر زیرنویس¹ که با دو براکت باز و بسته [] نمایش داده می‌شود استفاده می‌کنیم. بنابراین i امین عنصر دنباله v_1, \dots, v_n را به صورت $v[i]$ نمایش می‌دهیم.

¹ subscript

- الگوریتم‌ها در زمینه‌های زیاد و متنوعی کاربرد دارند.
- به عنوان مثال در پروژه ژنوم‌های انسانی هدف پیدا کردن الگوهای ژن‌ها در دی‌ان‌ای¹ انسان است که برای این کار از الگوریتم‌های کامپیوتری استفاده می‌شود. به عنوان چند مثال دیگر می‌توان از الگوریتم کوتاه‌ترین مسیر برای مسیریابی بسته‌های اینترنتی در شبکه‌های کامپیوتری، الگوریتم‌های رمزنگاری برای تبادل امن اطلاعات، الگوریتم‌های تخصیص منابع و زمانبندی در کاربردهایی مانند زمانبندی پروازها و تخصیص خلبان و خدمه به هواپیماها با کمترین هزینه ممکن و الگوریتم‌های فشرده سازی داده‌ها نام برد.
- معمولاً یک مسئله محاسباتی راه حل‌های زیادی دارد که بنابر معیارهای مورد اهمیت برای استفاده کننده الگوریتم، الگوریتمی انتخاب می‌شود که در یک یا چند معیار مورد نظر بهترین باشد. برای مثال در یک سامانه حمل و نقل هرچه به ازای یک سفر مسیر کوتاه‌تری طی شود، هزینه پایین می‌آید.

¹ DNA

- الگوریتمی بنویسید که عدد n را دریافت کند و مجموع $1 + 2 + \dots + n$ را چاپ کند.

الگوریتم

۱. عدد n را دریافت کن.
۲. اگر $n < 1$ ، برو به ۷
۳. $sum = 0$ ، $num = 1$
۴. $sum = sum + num$
۵. $num = num + 1$
۶. اگر $num > n$ برو به ۸ در غیر این صورت برو به ۴
۷. چاپ کن عدد ورودی غیر معتبر است. پایان.
۸. مقدار sum را چاپ کن. پایان.

- خط ۴ در این الگوریتم چند بار تکرار می شود؟

- خطوط ۴ و ۵ و ۶ در این الگوریتم n بار تکرار می‌شوند.
- می‌گوییم این الگوریتم از مرتبه n است، بدین معنی که اگر عدد n به الگوریتم داده شود، عمل محاسباتی جمع باید n بار تکرار شود. اگر هر عمل محاسباتی در زمان ۱ میلی ثانیه انجام شود، این الگوریتم در n میلی ثانیه محاسبات را انجام می‌دهد.
- آیا می‌توانید الگوریتم بهتری بنویسید که در زمان کمتری اجرا شود؟

۱. عدد n را دریافت کن.

۲. اگر $n < 1$ ، برو به ۴

۳. $sum = n (n + 1) / 2$

۴. چاپ کن عدد ورودی غیر متغیر است. پایان.

۵. مقدار sum را چاپ کن. پایان.

– الگوریتمی بنویسید که عدد n را دریافت کند و زوج و فرد بودن آن را تعیین کند.

۱. عدد n را دریافت کن.

۲. $r = n \% 2$

۳. اگر $r == 0$ برو به ۴ در غیر اینصورت برو به ۵

۴. چاپ کن عدد زوج است. پایان.

۵. چاپ کن عدد فرد است. پایان.

– الگوریتمی بنویسید که عدد n را دریافت کند و $n!$ را چاپ کند.

الگوریتم

۱. عدد n را دریافت کن.
۲. اگر $n < 0$ آنگاه برو به ۷
۳. $fact = 1$ ، $i = 1$
۴. $fact = fact * i$
۵. $i = i + 1$
۶. اگر $i > n$ آنگاه برو به ۸. در غیر اینصورت برو به ۴
۷. چاپ کن ورودی غیر معتبر است. پایان.
۸. مقدار $fact$ را چاپ کن. پایان.

- الگوریتمی بنویسید که عدد n را دریافت کرده و عدد فیبوناچی n ام را چاپ کند.

الگوریتم

۱. عدد n را دریافت کن.

۲. اگر $n < 0$ چاپ کن عدد ورودی غیر معتبر است. پایان.

۳. اگر $n \leq 1$ عدد n را چاپ کن. پایان.

۴. $i = 1$ ، $prev = 0$ ، $curr = 1$

۵. $next = prev + curr$

۶. $prev = curr$

۷. $curr = next$

۸. $i = i + 1$

۹. اگر $i < n$ برو به ۵

۱۰. عدد $curr$ را چاپ کن. پایان.

- الگوریتمی بنویسید که انتگرال $f(n)$ را به طور تقریبی از i تا j محاسبه کند.

۱. اعداد i و j را دریافت کن.

۲. $d = 1.0$ ، $sum = 0$

۳. $sum = sum + f(i) * d$

۴. $i = i + d$

۵. اگر $i > j$ آنگاه برو به ۶ در غیراینصورت برو به ۳

۶. مقدار sum را چاپ کن. پایان.

- الگوریتمی بنویسید که عدد n را دریافت کرده، و ارقام آن را از سمت راست به چپ چاپ کند.

۱. عدد n را دریافت کن.

۲. اگر $n < 0$ چاپ کن عدد ورودی غیر معتبر است. پایان.

۳. $r = n \% 10$

۴. $n = n / 10$

۵. رقم r را چاپ کن.

۶. اگر $n > 0$ آنگاه برو به ۳.

۷. پایان.

- الگوریتمی بنویسید که عدد n را دریافت کرده، آن را به مبنای ۲ تبدیل کند.

الگوریتم

۱. عدد n را دریافت کن.

۲. اگر $n < 0$ چاپ کن ورودی غیر معتبر است. پایان.

۳. $res = 0$ ، $base = 1$

۴. $r = n \% 2$

۵. $n = n / 2$

۶. $res = res + base * r$

۷. $base = base * 10$

۸. اگر $n > 0$ برو به ۴

۹. مقدار res را چاپ کن.

- الگوریتمی بنویسید که یک عدد دودویی را دریافت کرده، به مبنای 10 تبدیل کند.

الگوریتم

۱. عدد n را دریافت کن.

۲. $res = 0$ ، $base = 1$

۳. $r = n \% 10$

۴. $n = n / 10$

۵. $res = res + base * r$

۶. $base = base * 2$

۷. اگر $n > 0$ برو به ۳.

۸. مقدار res را چاپ کن.

- الگوریتمی بنویسید که دو عدد را دریافت کرده، کوچک‌ترین مضرب مشترک آنها را محاسبه کند.

۱. دو عدد m و n را دریافت کن.

۲. اگر $m \leq 0$ یا $n \leq 0$ چاپ کن ورودی غیر معتبر است. پایان.

۳. بزرگ‌ترین مقسوم‌علیه مشترک m و n را با استفاده از الگوریتم ب.م.م محاسبه کن و برابر با \gcd قرار بده.

$$4. \quad lcm = (n * m) / gcd$$

۵. مقدار lcm را چاپ کن.

۱. دو عدد m و n را دریافت کن.
۲. اگر $m \leq 0$ یا $n \leq 0$ چاپ کن ورودی غیر معتبر است. پایان.
۳. اگر $n > m$ آنگاه m و n را جابجا کن ($k=m$; $m=n$; $n=k$).
۴. $d = 1$
۵. $lcm = d * m$
۶. $d = d + 1$
۷. اگر $lcm \% n \neq 0$ آنگاه برو به ۵.
۸. مقدار lcm را چاپ کن.

- الگوریتمی بنویسید که یک عدد را دریافت کرده بررسی کند آن عدد اول است یا خیر.

الگوریتم

۱. عدد n را دریافت کن.

۲. اگر $n \leq 1$ چاپ کن عدد ورودی غیر معتبر است. اگر $n == 2$ برو به ۷.

۳. $d = 2$

۴. $r = n \% d$

۵. $d = d + 1$

۶. اگر $r == 0$ برو به ۸ در غیر این صورت اگر $n/2 \leq d$ برو به ۴.

۷. عدد n اول است. پایان.

۸. عدد n اول نیست. پایان.

- الگوریتمی بنویسید که مجموع دنباله $1 + 1/2 + 1/4 + 1/8 + 1/16 + \dots$ را محاسبه کند.

۱. $\text{sum}=0.0$; $d=1.0$; $t=0.0001$

۲. $\text{sum} = \text{sum} + d$

۳. $d = d/2$

۴. اگر $d > t$ برو به ۲

۵. مقدار sum را چاپ کن. پایان.

- الگوریتمی بنویسید که یک دنباله از اعداد را دریافت کرده، آنها را مرتب کند.

۱. عدد n (تعداد اعداد دنباله) را دریافت کن.

۲. $k = 1$

۳. عدد $A[k]$ را دریافت کن.

۴. $k = k + 1$

۵. اگر $k \leq n$ برو به ۳

$$۶. i = 1$$

$$۷. j = i + 1$$

۸. اگر $A[i] > A[j]$ مقدار $A[i]$ و $A[j]$ را جابجا کن ($k=A[i]; A[i]=A[j]; A[j]=k$).

$$۹. j = j + 1$$

۱۰. اگر $j \leq n$ برو به ۸

$$۱۱. i = i + 1$$

۱۲. اگر $i < n$ برو به ۷

$$۱۳. k = 1$$

۱۴. عدد $A[k]$ را چاپ کن.

$$۱۵. k = k + 1$$

۱۶. اگر $k \leq n$ برو به ۱۴.