

THE HULK

شناخت مسایل و راه ارایه راه حل مناسب برای حل آن ها
به نام او که هرچه هست از اوست



Author : Mehdi Abdollahzadeh

Contact: Hulk.Iran@Gmail.Com

My Id : [X_TheHulk_X](#)

ملاحظات:

شما دوستان عزیز من سعی دارم طی نوشتن چندین جزودر مورد الگوریتم این مبحث و فلوجارت همراه زبان Vb را با این مهارت بیشتر آشنا کنم. البته بسیار گسترده بود که من در اینجا سعی دارم بطور مختصر (البته زیادم مختصر نیست) و مفید توضیح بدم. این مقاله شامل توضیحاتی کامل در مورد الگوریتم و فلوجارت می باشد.

ملاحظات:

استفاده از مطالب این مقاله با ذکر نام نویسنده و گروههای مربوطه بلامانع میباشد.

WWW.STELLAR.BLOGFA.COM

© All Rights Reserved For **Hulk**

® All Rights Reserved For **STELLAR**

www.IranStars.com

THE HULK

شناخت مسایل و راه‌ارایه راه حل مناسب برای حل آن‌ها

کلیات:

روش حل مسایل از روش‌ها و تحلیل‌های ریاضی و منطقی اولین بار به وسیله دانشمند بزرگ ایرانی - خوارزمی - مورد توجه قرار گرفت که در علوم ریاضی و نجوم طرح کرد پایه اساس روشی را در حل مسایل بنیان گذاشت که در آینده مدتها مورد استفاده برنامه‌نویسان کامپیوتر قرار گرفت. نام الگوریتم نیز از عبارت "الخوارزمی" گرفته شده است. (خدا رو شکر) ☺

در این پیمان‌ه مهارتی با نحوه تجزیه و تحلیل انواع مختلف مسایل و جستجو و طراحی راه‌حلهای مناسب برای آنها آشنا خواهید شد و در پایان ارایه راه حل به دست آمده با توجه به قواعد علم الگوریتم را فرا خواهید گرفت.

۱- شناخت مسایل و راه‌ارایه راه حل مناسب برای حل آن‌ها

به منظور ارایه راه حل مناسب برای یک مساله، بهترین کار بررسی آن یا به عبارت دیگر تجزیه و تحلیل مساله است. به این منظور و برای شناخت بهتر یک مساله (مجهولات) و عملیات محاسباتی در این جا این عوامل عبارتند از:

مقادیر معلوم، **خواسته‌های مساله (مجهولات)** و **عملیات محاسباتی**. در این جا این پارامترها را در مورد بررسی قرار می‌دهیم.

مقادیر معلوم:

مقادیری که در اختیار مساله قرار می‌گیرند و برای رسیدن به هدف مورد نظر در مساله مورد نیاز هستند.

محاسبات:

برای رسیدن به نتایج مورد نظر معمولاً لازم است تا عملیاتی را روی مقادیر معلوم انجام دهید؛ بخش عمده‌ای از این عملیات معمولاً با استفاده از فرمول‌های مختلف انجام می‌شود؛ البته محاسبات می‌توانند با توجه به روابط منطقی که بین مقادیر معلوم و خواسته‌های مساله وجود دارند، انجام گیرند.

خواسته‌های مساله (مجهولات):

مقادیری هستند که معمولاً در اثر انجام عملیات روی مقادیر معلوم حاصل می‌شوند، البته مجهولات می‌توانند از روابط منطقی که در حل مساله دخالت می‌کنند نیز به وجود آمده و مورد استفاده قرار گیرند.

THE HULK

شناخت مسایل و راه ارایه راه حل مناسب برای حل آن ها

له عنوان مثال فرض کنید میخواهیم محیط و مساحت یک دایره به شعاع دلخواه را به محاسبه کنیم. برای حل این مسأله و ارائه راه حل مناسب با روش ارایه شده، ابتدا مقادیر معلوم برای حل این مسأله را مورد توجه قرار می دهیم؛ برای همان طور که می دانید برای محاسبه محیط و مساحت هر دایره باید شعاع آن را در اختیار داشته باشیم، بنابراین شعاع دایره (R) به عنوان تنها داده مورد نیاز برای حل مسأله کافی است.

محاسباتی که برای رسیدن به محیط و مساحت دایره لازم است، در واقع فرمول های زیر خواهند بود:

$$P=2*3/14*R$$

$$S=3/14*R^2$$

همان طور که می بینید با استفاده از این فرمول ها روابط بین داده ها ورودی و نتایج مورد نظر به خوبی تعیین می شود و بلاخره به عنوان مرحله آخر، همان گونه که صورت مسأله و روابط ریاضی ارایه شده برداشت می شود خوسته های مسأله (مجهولات) همان محیط (P) و مساحت (S) دایره هستند که در متغیر مربوطه ذخیره می شوند.

۲-۱ الگوریتم

پس از بررسی مسأله و شناخت سه عامل اصلی برای حل آن لازم است تا روشی برای حل مسأله ارایه شود. این روشهای علمی با استفاده تجارب قبلی در حل مسایل و به دلخواه کسی که مسأله را حل می کند به دست آید یا با استفاده از روش های علمی مبتنی بر تحلیل های ریاضی و منطقی صورت پذیرد. الگوریتم در واقع مفهومی است که مسایل را با استفاده از تحلیل های ریاضی و منطقی مورد بررسی قرار داده و راه حل مناسبی برای آن ارایه می کند.

۱-۲-۱ تعریف الگوریتم

به مجموعه ای از دستورالعملها که با زبان دقیق و قابل فهم به همراه جزییات لازم و به صورت مرحله به گونه ای اجرا شده که هدف خاصی (حل مسأله) را دنبال کنند و شروع و خاتمه آنها نیز مشخص باشد، الگوریتم می گویند.

به عبارتی دیگری می توان الگوریتم را به یک ماشین تشبیه کرد که مقادیر معلوم را دریافت کرده، روی آنها محاسباتی را انجام می دهد و در پایان خوسته های مسأله (مجهولات) را ارایه می کند.

THE HULK

شناخت مسایل و راه ارایه راه حل مناسب برای حل آن ها

همان طور که می بینید رابطه نزدیکی بین مفهوم الگوریتم و نحوه کار کامپیوتر در حل مسایل وجود دارد. بنابراین با استفاده از روش الگوریتم می توانید حل مسایل را به گونه ای طراحی کنید که برای تبدیل به زبان کامپیوتر نیز قابل فهم باشد. به طور معمول با مفهوم الگوریتم آشنا هستید و از آن استفاده می کنید، به عنوان مثال وقتی هر روز صبح برای به مدرسه رفتن آماده می شوید، اعمالی را به ترتیب و به صورت دقیق انجام می دهید. یعنی ابتدا از خواب بیدار می شوید، دست و صورت خود را می شوید، صبحانه می خورید و سپس لباس مناسب به تن می کنید و بعد از برداشتن وسایل و کتابهای مورد نیاز از در خانه خارج می شوید مسافتی را طی کرده و به مدرسه و کلاس خود می روید. اگر هر یک از این اعمال قبل از دیگری انجام دهید هدف مورد نظر که شکل درست به مدرسه رفتن است، انجام نمی شود. اکنون می توان این مراحل را به صورت زیر و به شکل خلاصه تر و قابل فهم تری بیان کرد:

- بیدار شدن از خواب
- شستن دست و صورت
- خوردن صبحانه
- پوشیدن لباس مدرسه
- برداشتن کتابهای درسی و دفتر و قلم
- خروج از خانه
- رفتن به مدرسه و ورود به کلاس درس

۲-۲-۱ شرایط الگوریتم

اکنون که با مفهوم الگوریتم آشنا شدید، لازم است ویژگی های یک الگوریتم را بشناسید.

الف - استفاده از زبان ساده، دقیق و قابل فهم

این ویژگی سبب می شود تا در انجام دستورالعمل ها همواره یک برداشت یکسان حاصل شود، در غیر این صورت برداشت های متفاوت سبب خواهد شد تا دستورالعمل ها نتایج متفاوتی را به همراه داشته باشند. زبان الگوریتم نیز می تواند یکی از زبان های گفتاری نوشتاری مانند فارسی، اینگلیسی و نظایر آن باشد.

THE HULK

شناخت مسایل و راه ارایه راه حل مناسب برای حل آن ها

ب - استفاده از جزییات کافی

این ویژگی سبب می شود تا دستورالعمل ها به طور کامل اجرا شوند و جود موارد نا مشخص یا ارایه دستورالعمل ها به صورت کلی و مبهم سبب مخدوش شدن نتایج خواهد شد.

ج - شروع و پایان الگوریتم

در یک الگوریتم باید شروع دستورالعمل ها مشخص باشد. هر الگوریتم یک نقطه شروع دارد که به عنوان اولین دستورالعمل از آن استفاده می شود، به علاوه پایان و خاتمه الگوریتم نیز باید تعیین شود. یک الگوریتم می تواند بیش از یک نقطه پایان داشته باشد.

د - ترتیب انجام دستورالعمل ها

یکی از ویژگی های مهم یک الگوریتم ترتیب اجرای دستورالعمل ها است ؛ اگر این کار به درستی انجام نشود ، پیش بینی نتیجه کار مشخص نخواهد بود. در یک الگوریتم ترتیب انجام عملیات با استفاده از شماره گذاری دستورالعمل ها از بالا به پایین انجام می شود که البته در صورت نیاز میتوان ترتیب اجرای دستورالعمل ها را نیز تغییر داد. در مباحث بعد به این مسأله خواهیم پرداخت.

نکته :

اگر در حل مسایل سه عامل اصلی را به دقت مشخص کنید، طراحی یک الگوریتم کار زیاد دشواری نخواهد بود. اگر مقادیر معلوم، خوسته های مسأله، فرمولها و روابط ریاضی و منطقی بین آنها را به درستی تعیین کنید ، نوشتن الگوریتم مختلف آسانتر خواهد شد.

هـ - جامع بودن

الگوریتم باید به شکلی طراحی شود تا با توجه به صورت مسأله و مفروضات آن در تمام حالت ها ، نتایج مناسب و صحیحی را ارایه کرده و در حالت های خاص یا داده های ورودی متفاوت ، نتایج درستی را ایجاد کند.

و - استفاده از حداقل دستورالعمل ها :

در یک الگوریتم باید از دستورات اضافه که سبب افزایش حجم الگوریتم می شود ، خودداری نمایید چرا که این کار الگوریتم را شلوغ کرده و باعث سردرگمی میشود.

۱-۲-۳ انواع دستورالعمل ها در الگوریتم

همان طور که میدانید الگوریتم مجموعه ای از دستورالعمل هاست . دستورالعمل ها انواع مختلفی دارند که در این جا به ذکر تعدادی از آن ها

THE HULK

شناخت مسایل و راه ارایه راه حل مناسب برای حل آن ها

می پردازیم :

- الف – دستورالعمل های ورودی
- ب – دستورالعمل های خروجی
- ج – دستورالعمل های محاسباتی
- د- دستورالعمل های شرطی
- ه – دستورالعمل های تکرار (حلقه ها)

الف – دستورالعمل های ورودی :

این دستورالعمل ها برای دریافت داده ها ورودی استفاده می شوند و معمولاً برای اجرای آنها از عباراتی مانند "بخوان ، دریافت کن یا بگیر" استفاده می شود.

ب – دستورالعمل های خروجی

این دستورالعمل ها برای نمایش اطلاعات خروجی یا پیام های مورد نیاز به منظور راهنمایی کاربر روی صفحه نمایش یا چاپ آن ها به وسیله چاپگر استفاده می شوند و معمولاً برای اجرای آن ها عباراتی مانند "نمایش بده ، چاپ کن" به کار می رود.

ج – دستورالعمل های محاسباتی

این دستورالعمل ها در واقع نحوه ارایه و استفاده از فرمول ها و انجام عملیات ریاضی و محاسباتی را تعیین می کنند و معمولاً برای اجرای آن ها از همان شکلی که در ریاضیات وجود دارد ، استفاده می شود یعنی در سمت راست تساوی عملیات محاسباتی و در سمت چپ تساوی نام یک متغیر به کار می رود، البته به جای علامت تساوی از علامت فلش (←) نیز استفاده می شود.

تعریف متغیر:

متغیرها مکانهایی هستند که توانایی نگه داری و ذخیره سازی انواع داده ها را دارند. متغیرها در الگوریتم همان کاربرد ریاضی را دارا هستند و علاوه بر اعداد می توانند متن یا مقادیر منطقی درست یا نادرست ، تاریخ ، ساعت و نظایر آن ها را نگه داری کنند.

د- دستورالعمل های شرطی

THE HULK

شناخت مسایل و راه ارایه راه حل مناسب برای حل آن ها

گاهی اوقات لازم است با مقایسه مقادیر داده ها ورودی ،اطلاعات خروجی ،متغیرها و غیره عملیاتی را در الگوریتم هدایت کرده و دستورالعمل های خاصی را اجرا کنید یا روند اجرای الگوریتم را با اتخاذ تصمیمات مناسبی کنترل نمایید.در این صورت از دستورات شرطی استفاده کنید؛شکل کلی یک دستورالعمل شرطی به یکی از این حالت هاست:

If	شرط(ها)	Then	دستور(ات)
If	شرط(ها)	Then	دستور(ات) Else

در حالت اول:

ابتدا شرط های ارایه شده بررسی می شوند و در صورتی که نتیجه بررسی درست باشد،دستور یا دستورات پس از **Then** اجرا می شوند،در غیر این صورت (نا درست بودن شرط بررسی شده) دستورالعملی که پس از دستورالعمل شرطی قرار گرفته ،اجرا خواهد شد بدون آن که دستور یا دستورات پس از **Then** اجرا شود.

در حالت دوم :

شکل کاملتری از دستورالعمل شرطی را ملاحظه می کنید که در ابتدا شرط یا شرط ها مورد بررسی قرار می گیرند؛ اگر نتیجه ارزیابی آن ها درست باشد،دستورات موجود در بخش **Then** اجرا می شوند. سپس دستورالعملی که پس از دستورالعمل شرطی قرار گرفته است، اجرا می شوند اما اگر نتیجه ارزیابی شرط یا شرط ها نادرست باشد بدون آن که دستورات بخش **Then** اجرا شوند،دستورات موجود در بخش **Else** اجرا خواهد شد سپس دستورالعملی که پس از دستور شرطی قرار دارد،اجرا می شود.

عمگرهای مقایسه ای

همان طور که گفته شد گاهی لازم است تا از دستورالعمل های شرطی در الگوریتم استفاده کنید.این گونه دستورالعمل ها ، عملیاتی را براساس نتیجه ارزیابی شرط یا شرط هایی که پس از عبارت اگر قرار داده می شوند،اجرا کرده یا از اجرای آن ها جلوگیری می کنند. برای طراحی و ایجاد شرط های

THE HULK

شناخت مسایل و راه ارایه راه حل مناسب برای حل آن ها

مورد نظر، از عملگرهای مقایسه ای استفاده می شود؛ قبلا از این گونه عملگرها در مباحث ریاضی استفاده کرده اید.

نکته:

در صورت نیاز، می توان یک دستورالعمل شرطی را در بخش دستورات بعد از **Then** یا **Else** دستورالعمل شرطی دیگری قرار داد؛ در این حالت به مجموعه دستورالعمل های شرطی دستورالعمل های شرطی تودرتو نیز میگویند.

عملگرهای منطقی

علاوه بر عملگرهای ریاضی و مقایسه ای، نوع دیگری از عملگرها به نوع عملگرهای منطقی نیز وجود دارند که برای ترکیب چند شرط و مقایسه آن ها با هم مورد استفاده قرار می گیرند.

عملگر	مفهوم
AND	"و" منطقی
OR	"یا" منطقی
NOT	نقیض

وقتی دو یا چند شرط با عملگر منطقی **AND** با هم ترکیب می شوند، نتیجه مقایسه، زمانی **True** خواهد بود که نتیجه تمام شرط ها و مقایسه **True** باشند، بنابراین اگر عبارت مقایسه ای **A** و **B** را در نظر بگیرید، جدول درستی عملگر منطقی **AND** به شکل زیر میباشد.

A	B	A AND B
T	F	F
F	T	F
F	F	F
T	T	T

در صورتی که دو یا چند شرط یا مقایسه را با عملگر منطقی **OR** با هم ترکیب کنید، نتیجه مقایسه زمانی **False** (نادرست) خواهد بود که نتیجه تمام شرط ها و مقایسه ها **False** (نادرست) باشد و در سایر حالات، نتیجه

THE HULK

شناخت مسایل و راه ارایه راه حل مناسب برای حل آن ها

True (درست) خواهد بود، در سایر حالات ،نتیجه True (درست) خواهد بود؛
جدول زیر جدول درستی عملگر منطقی OR خواهد بود.

A	B	A OR B
T	F	T
F	T	T
F	F	F
T	T	T

هـ - دستور العمل های تکرار (حلقه ها)

استفاده از دستور العمل هایی که تاکنون فرا گرفته اید در حل بعضی مسایل کافی نیست و بعضی از اوقات لازم است تا برخی از دستور العمل ها را به دفعات تکرار نمایید، در این موارد از دستور العمل تکرار یا همان حلقه ها استفاده کنید. دستور العمل حلقه از اجزای مختلفی تشکیل می شود که عبارتند از :

شمارنده حلقه :

یک متغیر عددی است که دفعات تکرار دستور العمل ها را در حلقه کنترل می کند. مقدار شمارنده در هر بار اجرای حلقه افزایش یا کاهش می یابد.

مقدار اولیه :

مقدار اولیه حلقه قبل از شروع حلقه تعیین می شود و به وسیله آن می توان مقدار اولیه را برای شمارنده حلقه تعیین کرد.

شرط حلقه :

برای کنترل تعداد دفعات تکرار حلقه ،باید از یک شرط استفاده کرد .شرط موجود در حلقه ،نقطه پایان تکرار دستور العمل ها را در حلقه مشخص می کند و باید به گونه ای تنظیم شود تا از ایجاد حلقه نا محدود جلوگیری کند .برای ایجاد شرط در یک حلقه می توان از دستور العمل های شرطی استفاده کرد.

دستورات حلقه :

بخش دیگری در حلقه ، دستور العمل هایی هستند که در داخل حلقه تکرار می شوند. این دستور العمل ها با توجه به نیاز مسأله انتخاب می شوند.

۱-۳ آرایه

الگوریتم هایی که تا کنون طراحی کرده اید داده ها را دریافت کرده و سپس بلافاصله مورد پردازش قرار می داد اما گاهی اوقات لازم است تا پردازش پس از ورود تمام یا بخشی از اطلاعات انجام شود یا عملیاتی چون مرتب سازی یا

THE HULK

شناخت مسایل و راه آرایه راه حل مناسب برای حل آن ها

جستجوی اطلاعات در داده های وارد شده انجام گیرد. در مواردی هم حم داده ها ورودی زیاد است و برای ذخیره سازی آن ها باید از متغیرهای متعددی با نام های متفاوت استفاده کرد و سپس از آن ها در پردازش بهره برد که کاری دشوار و در اکثر موارد غیر ممکن خواهد بود؛ به عنوان مثال فرض کنید می خواهید الگوریتمی بنویسید که داده های مربوط به مشخصات و نمرات ۱۰۰ دانش آموزیک مدرسه را ذخیره و پردازش نماید، تعریف متغیرهای مورد نیاز به این تعداد عملاً کار بسیار دشواری است. در چنین مواردی می توانید از یک مفهوم مهم که در ریاضیات با آن به شکل های مختلفی آشنا شده اید، استفاده کنید: آرایه، انجام این گونه عملیات را بسیار آسان و سریع می کند. آرایه مکان هایی از حافظه است که با یک نام شناسایی شده و دارای اندیس هایی (حلقه) برای اشاره به حلقه های این آرایه است. در نتیجه آرایه را می توان به صورت یک متغیر با مکان های متعددی که داده های هم نوع را در خود نگه می دارند، تعریف کرد. آدرس خانه های آرایه متوالی بوده و در حافظه به صورت پشت سر هم قرار میگیرند. برای درک بهتر این نحوه ذخیره سازی در یک آرایه به شکل زیر توجه کنید.

A1	A2	A3	A4	A5	...	An
----	----	----	----	----	-----	----

همان طور که مشاهده می کنید در شکل فوق یک آرایه با نام A و با تعداد n عضو ترسیم شده است و هر یک از فضاها به یکی از اعضای آرایه اختصاص دارد ولی تمام آن ها دارای نام A هستند که با استفاده از اندیس های ۱ تا n از یکدیگر متمایز می شوند. n یا تعداد اعضا به عهده شماست و می تواند یک مقدار ثابت یا متغیر باشد.

نکته:

دو آرایه را وقتی می توان جمع کرد که تعداد عناصر آن ها برابر باشد

۴-۱ زیر الگوریتم

در حل مسایل واقعی الگوریتم، معمولاً با تعدادی دستورالعمل که بارها در الگوریتم استفاده شده اند، مواجه می شوید که طراحی و نوشتن چنین الگوریتمی کار بسیار دشوار و زمان بری است. در چنین شرایطی بهتره است تا این بخش ها

THE HULK

شناخت مسایل و راه ارایه راه حل مناسب برای حل آن ها

به صورت جداگانه طراحی و نوشته شوند و در هر مرحله از الگوریتم مورد نظر مورد استفاده قرار گیرند. برای انجام این کار نیاز به مفهوم دیگری به نام زیر الگوریتم خواهید داشت. زیر الگوریتم در واقع یک الگوریتم است که به حل مسأله در الگوریتم اصلی کمک می کند. استفاده از زیر الگوریتم مزایای زیر را دارد:

- ۱- امکان تعمیم و ارتقای آسان تر در آینده
- ۲- جلوگیری از تکرار دستورالعمل
- ۳- خطایابی و خطا زدایی آسان تر و سریع تر
- ۴- استفاده از زیر الگوریتم در الگوریتم مشابه

زیر الگوریتم

- ۱- شروع
- ۲- $J \leftarrow I$
- ۳- $FACT \leftarrow FACT * J$
- ۴- $J \leftarrow J - 1$
- ۵- اگر $1 \geq J$ آن گاه برو به مرحله ۳
- ۶- بازگشت

الگوریتم اصلی

- ۱- شروع
- ۲- X و N را دریافت کن
- ۳- $J \leftarrow I$ و $K \leftarrow I$ و $SUM \leftarrow 0$
- ۴- $FACT \leftarrow 1$
- ۵- زیر الگوریتم (Fact, I) Factorial را اجرا کن
- ۶- $SUM \leftarrow SUM + K * \frac{X!}{Fact}$
- ۷- $I \leftarrow I + 2$
- ۸- $K \leftarrow K * -1$
- ۹- اگر $1 \leq 2n + 1$ آن گاه برو به مرحله ۴
- ۱۰- SUM را نمایش بده
- ۱۱- پایان