



2019

خرداد ۹۸

Jun 8

۱۴۴۰

شنبه

۴ شوال

مسائل محاسباتی

همانطور که پیش تر گفتیم، کار اصلی یک CS حل مسائل محاسباتی است. مسائل محاسباتی را از دید قابلیت حل پذیری به دو دسته زیر تقسیم می کنند:

- غیر قابل حل: که برای آن هیچ الگوریتمی (برنامه ای) نمی تواند

وجود داشته باشد. مسئله توقف

(Halting Problem)

- قابل حل: مسائلی که برای آن الگوریتمی (برنامه ای) وجود دارد.

مسائل قابل حل نیز در دو دسته قابل تقسیم بندی هستند:

- قابل حل در تنوع: در عمل نمی توان این برنامه ها

را برای نمونه های بزرگ به

کار گرفت. مسئله مسیر

در مسافتی

- قابل حل در عمل: در عمل برنامه ها این

کلاس قابل اعتقاد هستند

مگر در صورتی که تعداد

الگوریتم

یک الگوریتم مجموعه ای از اعمال گام به گام است که هدف مشخصی را دنبال می کند و

به طور کلی دارای ویژگی های زیر است:

۱. ورودی (Input):

۲ خروجی (Output)

۳ عمومیت (Generality): قابل اعمال روی مجموعه‌ای از ورودی‌ها است،

۴ منتهی (Finiteness): بعد از تعداد گام‌های متناهی متوقف می‌شود.

۵ دقت (Precision): گام‌ها به طور دقیق تعریف شده‌اند.

۶ صحت یا درست‌نمایی (Correctness): خروجی تولید شده توسط الگوریتم درست است.

مثال: ① انجام محاسبه جمع دو عدد
② انجام محاسبه ^{تفریق دو عدد} ~~حاصل~~ ^{درخواست} ~~محاسبه~~ ^{درخواست}

③ انجام محاسبه یک عمل بر روی دو عدد ^{درخواست}

الگوریتم یک چارچوب یا قالب برای بیان مسأله حل شده است و نحوه نمایش آن می‌تواند به شکل از صورت‌های زیر باشد:

۱ بیان در قالب متن فارسی یا انگلیسی، ^{آهسته}
۲ بیان در قالب نمودارها، ^{رسمی}
۳ بیان در قالب شبه‌کد،

۴ بیان در قالب یک برنامه در یک زبان برنامه‌نویسی خاص

و ...

2019

۹۸ خرداد

Jun 10



۱۴۴۰

دوشنبه

۶ شوال

مثال ۱: در قالب متن فارسی، کاسب جمع دو عدد را خوا

گام ۱: شروع،

گام ۲: عدد را خوا اول را دریافت کن،

گام ۳: عدد را خوا دوم را دریافت کن،

گام ۴: جمع دو عدد دریافت شده را کاسب کن،

گام ۵: حاصل جمع را اعلام کن،

گام ۶: پایان.

Example 1. Addition of two arbitrary numbers:

step1. start

step2. Take the first number,

step3. Take the second number,

step4. compute the sum of the two received numbers,

step5. Announce the result of step4,

step6. End

در ادامه مقدماتی بیان الگوریتم را در قالب سئوالاتی سبب به مثال هار حل می کنیم.

به منظور سه ساختارهای بار نوشتن برنامه، به یکد، مقدماتی و جود دارد:

۱ دنباله (sequence)

۲ انتخاب (Branching) که انتخاب (selection) هم گفته می شود.

۳ حلقه (تکرار) Loop
↓
Repetition

۱ دنباله یا sequence: کوالر از دستور ها که یکی پس از دیگری می آیند:

- خواندن از ورودی: Read

- نوشتن در خروجی: Print

- محاسبات: محاسبات ریاضی، منطقی، انتخاب

مثال: خواندن یک عدد و چاپ دو برابر آن در خروجی:

1. Start

2. Read A

3. $C = A * 2$

4. Print C

5. End

انتخاب / Branching: یک تصمیم دودویی بر پایه یک شرط است.
condition

از عبارت زیر در کامپایلر انتخاب می‌کنیم.

دستور then شرط If

~~else~~

~~و نه~~

مثال: دو عدد از ورودی دریافت می‌کنند و بزرگترین

آنها را در خروجی چاپ می‌کند.

~~End~~

1. Start

2. Read A, B

3. If A greater than or equal to B then $C = A$

2019

خرداد ۹۸

Jun 13

۱۴۴۰
پنجشنبه
۹ شوال

4. If B greater than A then $C = B$

5. print C

6. End

حل: حلقه تکرار می‌شود. یک عدد صحیح از ورودی دریافت و قدر مطلق آن را در خروجی چاپ کنید.
* سه عدد از ورودی دریافت و بزرگترین آن‌ها را در خروجی چاپ کنید.

حلقه (Loop): اجازه می‌دهد که یک دستور یا توالی از دستورات به تکرار اجرا شود.
به طور مکرر اجرا شود.

نمونه اجرای حلقه از قالب زیر استفاده می‌کنیم:

If شرط then go to line #n

جاییکه #n به بزرگترین شماره خط قبل از خط جاری است.

2019

خرداد ۹۸

Jun 15



۱۴۴۰

شنبه

۱۱ شوال

مثال: چاپ اعداد زوج از ۲۰ تا ۹۹:

1. Start

2. $I \leftarrow 0$

3. print I

4. $I = I + 2$

5. IF ($I \leq 98$) then goto line 3

6. End

تمرین: ۱) چاپ اعداد فرد طبیعی که کوچکتر مساوی n (جائگه n از ورودی دریافت می شود)،
۲) چاپ اعداد زوج بین 2000 تا 20000 و چاپ جمع کل اعداد چاپ شده.