

C++

پریسا حامد روح بخش

موسسه ی پارس پژوهان

فصل اول

- C++ ؟
- نصب الزامات
- چی بخونیم؟
- مقدمات پایه زبان C++
- الگوریتم
- مراحل حل مسئله

C++ ؟

- C++ یک نسخه پیشرفته و توسعه یافته از زبان برنامه نویسی C است که توسط Bjarne Stroustrup در سال 1979 توسعه یافت. این زبان در ابتدا C with Classes نام داشت و بعداً به C++ تغییر نام داد. Bjarne Stroustrup از C برای ساخت آنچه می خواست استفاده کرد زیرا C یک زبان همه منظوره کارآمد و سریع بود.
- در مقایسه با زبان C ویژگی های بسیار جدیدی را معرفی نموده است.



- آرگومان پیش فرض
- وراثت
- توابع مجازی
- فضاها و نام و اشاره گر ها و ...

C++ ؟



• ویژگی های مهم زبان C++ :

(1) شی گرا

(2) سطح بالا

(اما به طور کلی به آن یک زبان سطح میانی می گویند چرا که هم قابلیت های یک زبان سطح بالا را دارد و هم سطح پایین). این زبان برنامه نویسی بسیار قدرتمند است و از خانواده C محسوب میشود.

(3) سریع و قدرتمند

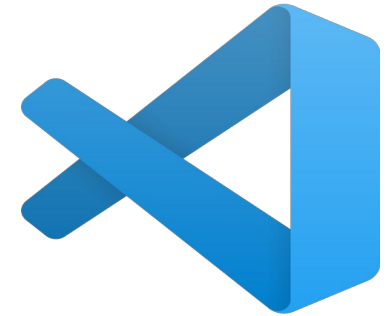
(4) مدیریت حافظه در اختیار برنامه نویس

C++ ؟

- C++ اولین بار در سال 1985 منتشر شد، اما اولین نسخه استاندارد آن (C++98) در سال 1998 انتشار پیدا کرد. در سال 2003، C++03، با هدف رفع اشکالات شناسایی شده در C++98 و اطمینان از اینکه زبان قابلیت حمل و سازگاری بیشتری دارد، منتشر شد.
- ارتقاء عمده بعدی این زبان هشت سال بعد (2011) انجام شد و C++11 نام گرفت. پس از آن، C++14 منتشر شد و پیشرفت های ساده ای انجام داد و باگ های C++11 را رفع کرد. نسخه C++17 دارای چندین ویژگی جدید و کتابخانه استاندارد بسیار گسترده است و در نهایت نسخه فعلی که در سال 2020 منتشر شد C++20 نامیده می شود.

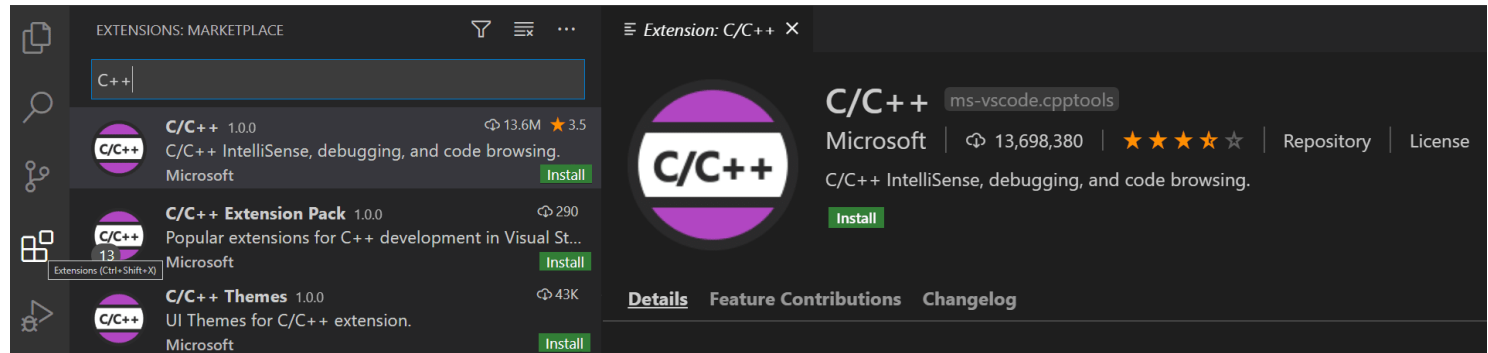
چی نصب کنیم؟

- Visual Studio Code == VSCode
- For Windows , Linux and Mac



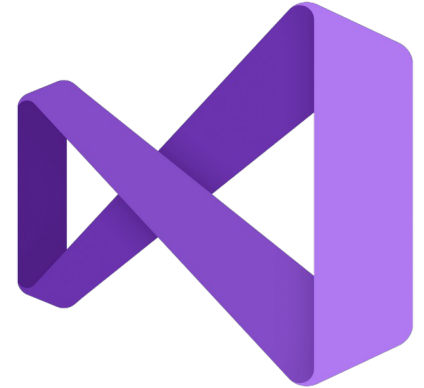
<https://code.visualstudio.com/download>

<https://code.visualstudio.com/docs/cpp/config-linux>



چی نصب کنیم؟

- Microsoft Visual Studio 2019
- [Installation link](#)



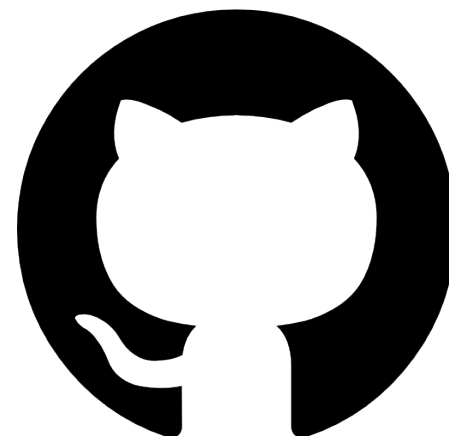
چی نصب کنیم؟

- QtCreator
- Qt 5.14.0 Win/Mac/Linux
- <https://www.qt.io/>



چی نصب کنیم؟

- GitHub
- <https://github.com/>



• دوره رایگان و مفید یادگیری گیت:

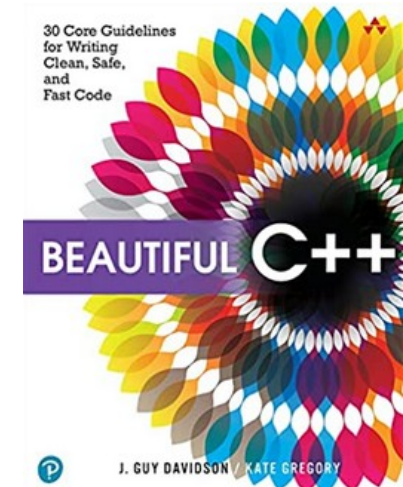
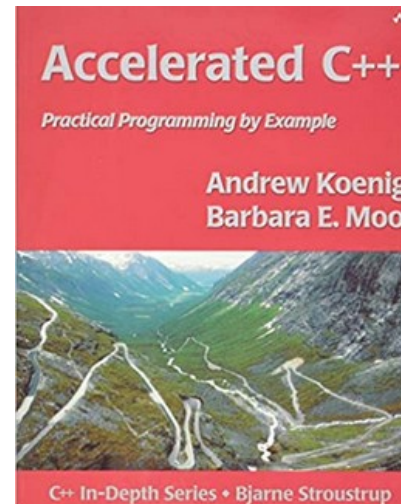
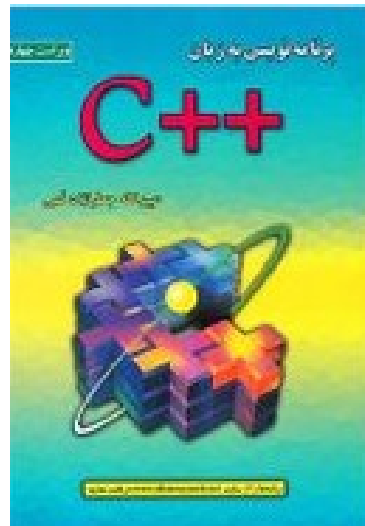
لینک دوره - زمان مورد نیاز حدود ۱ ساعت

چی بخونیم؟

• کتاب

• فارسی <== برنامه نویسی به زبان C++ (عین الله جعفر نژاد قمی)

- Accelerated C++: Practical Programming by Example (C++ In-Depth Series)
- Beautiful C++: 30 Core Guidelines for Writing Clean, Safe, and Fast Code



چی بخونیم؟

• سایت های خوب

- <https://www.cppstories.com/p/resources/>
- <https://github.com/rigtorp/awesome-modern-cpp>
- <https://www.geeksforgeeks.org/>
- <https://www.hackerrank.com/domains/cpp>

- **StackOverflow**

مقدمات پایه زبان C++


- A C++ program is a collection of commands or statements.


```
#include <iostream>
using namespace std;

int main()
{
    return 0;
}
```

**remember that the entry point of every C++
program is the main() function,
irrespective of what the program does.**

مقدمات پایه زبان C++

 **Curly brackets { }** indicate the beginning and end of a function, which can also be called the **function's body**. The information inside the brackets indicates what the function does when executed.

 In C++, the semicolon is used to terminate a statement. Each statement must end with a semicolon. It indicates the end of one logical expression.

مقدمات پایه زبان C++

CPP

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello world!";
7     cout << " This " << "is " << "awesome!";
8     return 0;
9 }
```

Hello world! This is awesome!

Headers and namespaces

- C++ offers various headers, each of which contains information needed for programs to work properly.
- We have already seen the standard `<iostream>` header on our first C++ program.
- `#include` is used for adding a standard or user-defined header files to the program.

The `<iostream>` header defines the standard stream objects that input and output data.

Headers and namespaces

- A **namespace** is a declarative region that provides a **scope** to the identifiers (names of elements) inside it.
- In our code, the line using namespace std; tells the compiler to use the **std** (standard) namespace



The **std** namespace includes features of the **C++ Standard Library**.

مقدمات پایه زبان C++

CPP

```
1 #include <iostream>
2 using namespace std;
3
4 int main()
5 {
6     cout << "Hello world!" << endl;
7     cout << "I love programming!";
8     return 0;
9 }
```

```
Hello world!
I love programming!
```



The **cout** object does not insert a line break at the end of the output.

One way to print two lines is to use the **endl** manipulator, which will put in a line break.

مقدمات پایه زبان C++

- The new line character `\n` can be used as an alternative to `endl`.
- Using a single `cout` statement with as many instances of `\n` as your program requires will print out multiple lines of text.

مقدمات پایه زبان C++

- **Comments**
- **Comments are explanatory statements that you can include in the C++ code to explain what the code is doing.**
- **The compiler ignores everything that appears in the comment, so none of that information shows in the result.**



Adding comments to your code is a good practice. It facilitates a clear understanding of the code for you and for others who read it.

مقدمات پایه زبان C++

مثلا

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      // prints "Hello world"
7      cout << "Hello world!";
8      return 0;
9  }
```

Hello world!

مقدمات پایه زبان C++

- **Multi-Line Comments**



Comments can be written anywhere, and can be repeated any number of times throughout the code.

- Comments that require multiple lines begin with `/*` and end with `*/`
- You can place them on the same line or insert one or more lines between them.

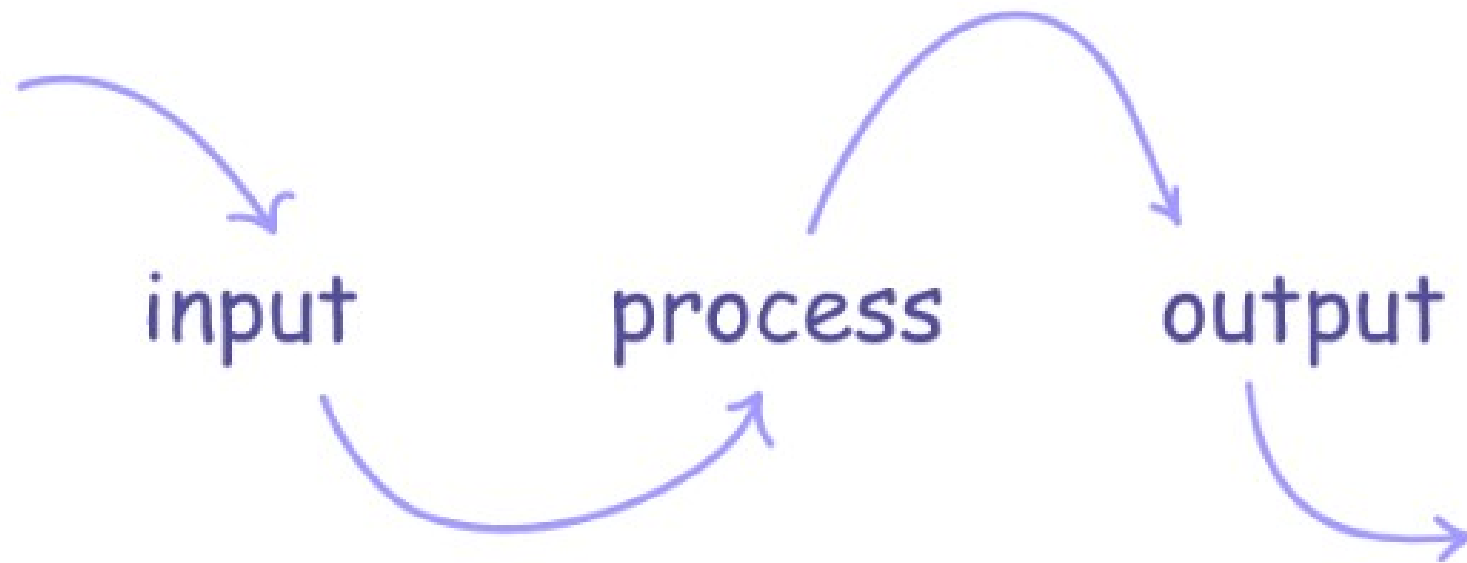
```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      /*We are going to
7       print "Hello world!" */
8
9       cout << "Hello world!"; // prints Hello world!
10
11      return 0;
12 }
```

الگوریتم چیست؟

- **الگوریتم** مجموعه‌ای متناهی از دستورالعمل‌ها است، که به ترتیب خاصی اجرا می‌شوند و مسئله‌ای را حل می‌کنند. به عبارت دیگر یک الگوریتم، روشی گام به گام برای حل مسئله است.
- واژه الگوریتم از نام «محمد بن موسی» خوارزمی گرفته شده است که ریاضیدانی ایرانی بود و در قرن سوم هجری زندگی می‌کرد. او به **خوارزمی** مشهور بود و واژه **Algorithm** به معنی **شیوه و روش محاسبه** از این نام گرفته شده است.



خصوصیات الگوریتم



خصوصیات الگوریتم

• ورودی

هر الگوریتم باید هیچ یا حداقل یک پارامتر را به عنوان ورودی دریافت کند. برای مثال ورودی الگوریتم می‌تواند تعدادی عدد، رشته‌ای از حروف، یک فایل متنی، یک معادله ریاضی و... باشد.

خصوصیات الگوریتم

• خروجی

هر الگوریتم باید حداقل یک کمیت به عنوان خروجی (نتیجه‌ی الگوریتم) تولید کند. خروجی الگوریتم معمولاً همان پاسخ مسئله‌ای است که الگوریتم برای تولید آن طراحی شده است. پس می‌تواند با توجه به درخواست مسئله به هر شکلی (تعدادی عدد، رشته‌ای از حروف، یک فایل متنی و...) باشد.

خصوصیات الگوریتم

• پردازش

الگوریتم باید دارای شروع و پایان مشخصی باشد، به نحوی که اگر دستورهای آن را دنبال کنیم، برای تمامی حالت‌های مختلف ورودی، الگوریتم پس از طی تعداد متناهی مرحله، خروجی مورد انتظار را تولید کند. دستورهای الگوریتم باید با زبانی دقیق و بی‌ابهام بیان شوند. هر دستور نیز باید انجام پذیر باشد.

- در روند پردازش الگوریتم، **زمان و حافظه مورد نیاز** برای دنبال کردن دستورها، از شروع تا پایان الگوریتم، باید به گونه‌ای **معقول و کم** باشد.

مراحل حل مسئله

- انسان برای افزایش سرعت و دقت در حل مسائل به استفاده از کامپیوترها روی آورده، اما معمولاً الگوریتم‌ها به شکلی که برای ما ساده و قابل فهم هستند، برای کامپیوتر معنا و مفهوم ندارند. برای مثال گفتن «به در نگاه کن و اگر در باز شد، سلام کن!» به یک کامپیوتر، برای اینکه این کار را انجام دهد کافی نیست. چرا که کامپیوتر باید دقیقاً بداند معنای «نگاه کردن» چیست، چگونه «باز شدن در» را تشخیص دهد و چگونه «سلام» کند. یک کامپیوتر فقط می‌تواند دستورالعمل‌هایی محدود به اعمال اولیه و تعریف شده‌ی خودش را اجرا کند.
- پس باید مسائل خود را به مدلی تبدیل کنیم و الگوریتم مورد نظر را با توجه به این مدل پیاده سازی کنیم. پس مراحل حل یک مسئله را می‌توانیم به ۳ قسمت زیر تقسیم کنیم

مراحل حل مسئله

mathematical model

algorithm

implementation



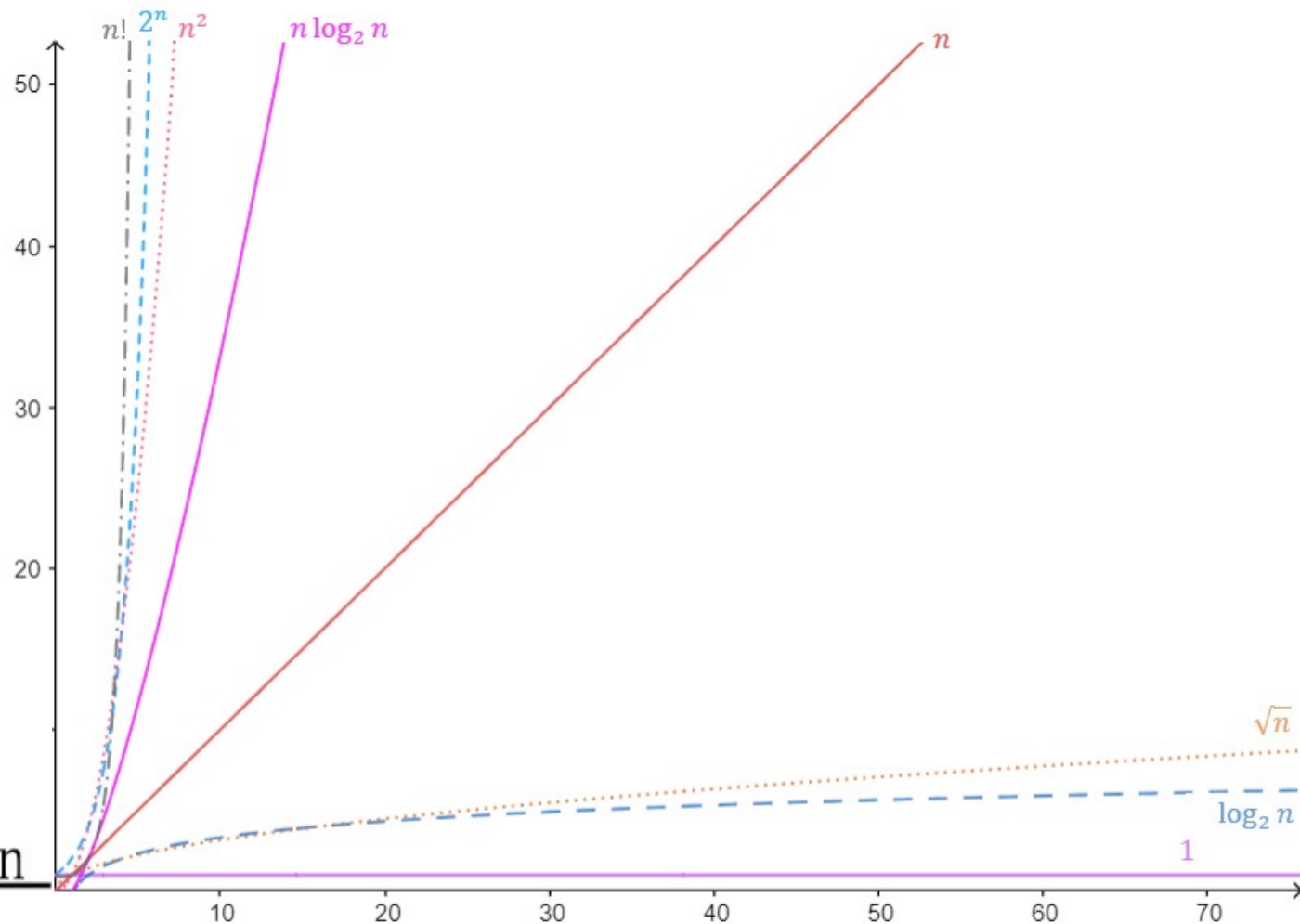
مدل سازی

- در اولین مرحله‌ی حل مسئله، باید آن را از **حالت ناواضح و شهودی** به **حالت واضح و دقیق تبدیل کنیم**. یکی از بهترین روش‌ها برای این کار استفاده از **مدل‌های ریاضی** است. برای مثال می‌توانید مسئله را به حل یک معادله و یا یک مسئله بهینه‌سازی تبدیل کنید. در ادامه خواهیم دید یکی از پر کاربردترین مدل‌ها **گراف‌ها** هستند و می‌توانیم مسائل مرتبط با «روابط دوستی در یک شبکه اجتماعی» و یا «جاده‌های بین شهرها» را با کمک گراف‌ها به یک مدل ریاضی تبدیل کنیم.

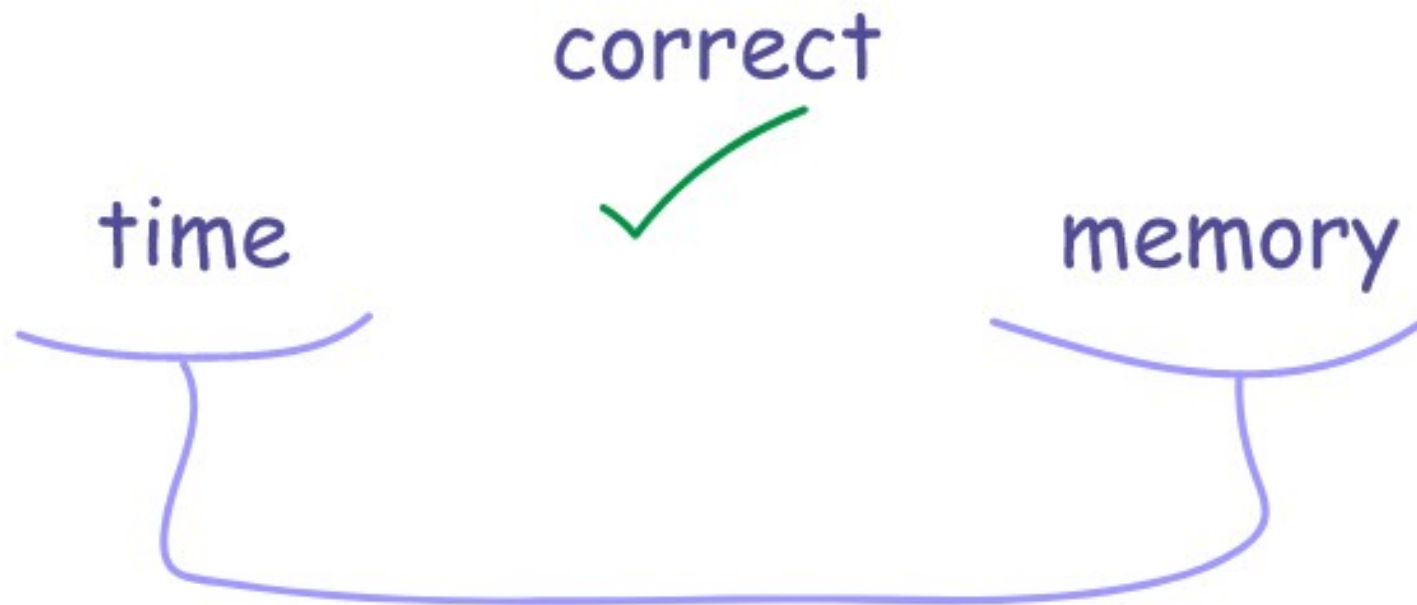
الگوریتم

- مرحله بعدی حل مسئله، ارائه یک **الگوریتم** برای بدست آوردن پاسخ مسئله باتوجه به **مدل واضح و دقیق** است. توجه کنید اگر الگوریتم‌ها را برای مدل‌های کلی ارائه دهیم، می‌توانیم همه مسائل قابل تبدیل به این مدل را حل کنیم.
- برای ارائه یک الگوریتم به صورت دقیق و قابل اجرا نیاز به **پیاده‌سازی** آن با یک **زبان برنامه‌نویسی** داریم. در واقع زبان‌های برنامه‌نویسی واسطه بین الگوریتم‌های طراحی شده توسط ما (انسان‌ها) و ماشین (کامپیوتر) است.

تحليل الگوریتم‌ها



معیار مقایسه الگوریتم‌ها





Question ?