Алгоритми. Видове алгоритми и подход при решаване на задачи.

Досега учихме главно синтаксис.

1. Какво е стойност?
2. Какво е израз?
3. Какво е променлива?
4. Какво е statement?
5. Какво е if statement?
6. Какво е while statement?
7. Потребителски вход
8. Списъци

Например:

```
books = ["Pragmatic Thinking and Learning",
         "Learn You a Haskell"]
```

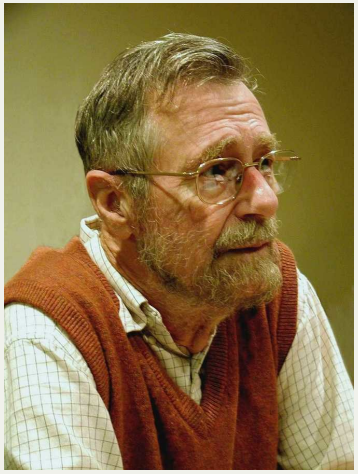
```
print(books[0])
```

```
a = 0
print(books[a])
```

```
print(books[len(books) - 1])
```

Алгоритъм.

Това представлява поредица от стъпки, чрез които се решава даден **проблем.**



Програмиране, проблеми и алгоритми.

1. Програмирането съществува, за да решава проблеми.
2. Проблемите се решават чрез алгоритми.
3. Алгоритмите нямат нужда от конкретен език за програмиране.
4. Те могат да се изразят като поредица от стъпки.

Примерен алгоритъм.

1. Кипнете водата до 100 градуса целзий.
2. Сложете чая във врялата вода.
3. Изчакайте 5 минути.
4. Махнете чая.
5. Сложете мед или захар или мляко.
6. Изпийте чая!

Друг алгоритъм.

1. За всеки човек, който дойде на входа.
2. Виж личната му карта.
3. Ако е >= на 18 години, го пусни.
4. Ако не е - помоли го да си тръгне.

Сумата на всички цели числа между 1 и N

1. Нека сумата S в началото да е = 0
2. За всяко число между 1 и N включително
3. S = S + конкретното число
4. Накрая имаме сумата в S.

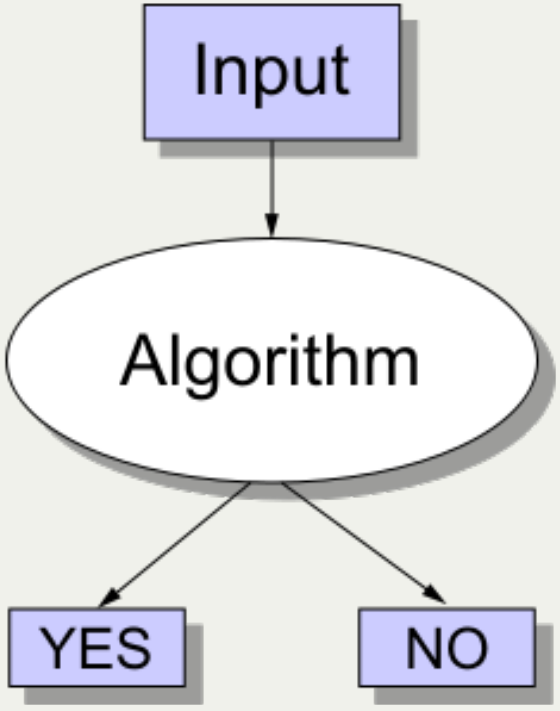
Един алгоритъм не се нужда от конкретен език за програмиране.

1. Ние го описваме в псевдо-код.
2. Имплементираме го в език за програмиране (например Python), за да го приложим в реалността.

Алгоритми, които отговарят с Да или Не.

1. Просто ли е дадено число?
2. Перфектно ли е дадено число?
3. Еднакви ли са два списъка?
4. Имаме ли път от град A до град B?
5. Квадратна ли е матрица?
6. Намира ли се елемент в даден списък?
7. Палиндром ли е даден низ?
8. И много други.

Yes/No алгоритъм може да се представи така:



Дали дадено число е просто?

1. Четем едно число N.

```
n = input("Enter a number: ")
n = int(n)

start = 2
is_prime = True

# Специален случай за 1.
if n == 1:
    is_prime = False

while start < n:
    if n % start == 0:
        is_prime = False
        break
    start = start + 1

if is_prime:
    print("It is prime")
else:
    print("It is not prime")
```

is_prime се нарича флагова променлива

1. В нея се пази крайният резултат от сметката.
2. Много често почва като True и ако нещо успее да я промени, става на False.
3. Много от Да/Не проблемите се решават по подобен начин.

Алгоритми, които броят нещо.

1. Колко прости числа има в интервал?
2. Колко перфектни числа има в интервал?
3. Колко са повтарящите се елементи в даден списък?
4. Колко са числата-палиндроми в даден интервал?
5. Колко са гласните в даден низ?
6. Колко са пътищата от град A до град B?

Да преброим простите числа в даден интервал.

1. Нека `counter = 0` - броят на простите числа в началото.
2. За всяко число **n** в интервала **[1, N]**
3. Ако **n е просто** - `counter += 1`
4. Накрая в `counter` ще имаме резултатът.

Много често решаваме проблем, като го разбием на по-малки проблеми, които сме решавали преди.

Ако знаем как да проверим дали число е просто, може да намерим простите числа в интервал.

Ако знаем как за **X** да проверим дали е изпълнено дадено **Да/Не условие**

=>

Може да кажем броя на всички **X** в интервал.

Решете задачите от 3-Simple-Algorithms от седмица 1