

ЛЕКЦИЯ 3 АЛГОРИТМИ

-  **Понятие за алгоритъм**
-  **Елементарни действия**
-  **Пример: приготвяне на кафе**
-  **Пример: алгоритъм на Евклид**
-  **Понятие за подалгоритъм**
-  **Комп. алгоритъм и програма**
-  **Свойства на комп. алгоритми**

ПРОГ_03

1/28

ПОНЯТИЕ ЗА АЛГОРИТЪМ

Римският принцип „*разделяй и владей*“ е добре известен от историята.

Той се прилага успешно когато искаме да изпълним или да обясним някое сложно действие, например *как да приготвим някакво ястие*.

Затова **понятието алгоритъм** **ИНТУИТИВНО** се свързва с понятия като **МНОЖЕСТВО ОТ ПРАВИЛА** (инструкции, команди и пр.), разпореждане, предписание, план, рецепта и др.

ПРОГ_03

2/28

ЕЛЕМЕНТАРНИ ДЕЙСТВИЯ

Един алгоритъм представя дадено
сложно действие чрез редица
от достатъчно прости (**елементарни**)
действия, които изпълняващите ги
могат да извършат **в** няколко
последователни стъпки и без да им са
необходими допълнителни обяснения.

ПРОГ_03

3/28

ОЩЕ ЗА ПОНЯТИЕТО

Алгоритъмът е **средство да се възложи**
определена дейност на изпълнител,
от който се изисква **само:**

- ① да може **да извършва** използваните
в алгоритъма **елементарни действия;**
- ② да **изпълнява тези действия**
в посочената от алгоритъма
последователност.

ПРОГ_03

4/28

СВЪРЗАНИ С АЛГОРИТЪМ СА:

- ① **съставител**: **подготвя** алгоритъма като **подбира** елементарните действия и **реда** на тяхното изпълнение;
- ② **изпълнител**: **изпълнява** предписаните елементарни действия **при конкретни начални условия**;
- ③ **потребител**: **задава** изпълнението на алгоритъма и **ползва крайния резултат**.

ПРОГ_03

5/28

НЕФОРМАЛНО ОПРЕДЕЛЕНИЕ

Описание на дискретен процес, който започва **от** определено **начално състояние** (**входна информация**) и достига **до резултат** (**изходна информация** – друго, крайно състояние).

Състоянията може да са **както реално** съществуващи, **така и** да представят **абстрактни модели** (данни, отношения).

ПРОГ_03

6/28

ИСТОРИЧЕСКИ СВЕДЕНИЯ

Строго определени **правила за достигане** на някаква **цел** са били формулирани **най-напред в математиката**. От дълбока древност например е известна система от правила (наречена **алгоритъм на Евклид**), определяща най-големия общ делител $\text{НОД}(a, b)$ на всеки две естествени числа a и b .

Идеята за съставяне (композиране) на алгоритми (и действия) от по-прости намира **широко приложение** в компютърната информатика.

Всеки алгоритъм моделира протичането на **процес** в природата, обществото или науката.

ПРОГ_03

7/28

ПРОИЗХОД НА ТЕРМИНА

Терминът **алгоритъм** произлиза **от** името на **Абу Джафар Мохамед ибн Муса ал-Хорезми** (арабски математик), който **около 820 г.** от н. е. написва научен **трактат за** това как да се представят (записват) числата в **10-ична бройна система** и как да се смята с тези представяния.

Съчинението на ал-Хорезми (**al-Gorezmi**) **е преведено** на латински (европейски) език в Средновековна Европа **в началото на XII-ти век**.

По-късно съдържанието на понятието се разширява – като **алгоритми** започват да се разглеждат и повечето известни по онова време **изчислителни процедури**.

ПРОГ_03

8/28

ПРИМЕР: АЛГОРИТЪМ за приготвяне на чаша кафе

Вход: 1 чаена лъжица кафе, 1 чаша вода, захар на вкус.

Изход: Чаша горещо кафе.

- ① Пригответе се за работа.
- ② Смесете всичко в подходящ съд.
- ③ Поставете на бавен огън.
- ④ Варете до кипване.
- ⑤ Налейте в чаша.
- ⑥ Поднесете горещо.
- ⑦ Прекратете работа.

ПРОГ_03

9/28

ПРИМЕР: АЛГОРИТЪМ на Евклид за определяне на НОД

Вход: Две естествени числа a и b .

Изход: НОД на a и b .

- ① Пригответе се за работа.
- ② Въведете и запомнете числата a и b .
- ③ Ако $a \neq b$, то изпълнете стъпка ④, иначе – ⑥.
- ④ Ако $a > b$, то изчислете $a - b$ и го помнете като a , иначе изчислете $b - a$ и го помнете като b .
- ⑤ Изпълнете стъпка ③.
- ⑥ Съобщете стойността на a (като резултат).
- ⑦ Прекратете работа.

ПРОГ_03

10/28

ПОНЯТИЕ ЗА ПОДАЛГОРИТЪМ

В примерите **първите** (и **последните**) **стъпки** **съвпадат** и освен това алгоритъмът е с **един вход** и **един изход**.

Това позволява той да бъде разглеждан като **ново**, по-сложно **действие** (с **определено начало и край**) и следователно да бъде **използван като част** от описанието на **друг алгоритъм**.

В такива случаи **първият алгоритъм се нарича подалгоритъм на втория**.

Чрез подалгоритми можем **да намалим сложността** на решаваната задача и **да се освободим** от ограничения парк **от елементарни действия** на изпълнителите.

ПРОГ_03

11/28

ПРИМЕР: ОПРЕДЕЛЯНЕ НА НОД НА ТРИ ЧИСЛА

Вход: Три естествени числа a , b и c .

Изход: НОД на a , b и c .

- 1 Пригответе се за работа.
- 2 Въведете и запомнете числата a , b и c .
- 3 Изчислете **НОД(a, b)** и го запомнете като d .
- 4 Изчислете **НОД(d, c)** и го запомнете като r .
- 5 Съобщете стойността на r (**резултат**).
- 6 Прекратете работа.

ПРОГ_03

12/28

ПОДАЛГОРИТЪМ НОД(a, b)

Получаваме: Две естествени числа a и b .

Връщаме: НОД на a и b .

- ① **Запомнете** от къде дойдохте тук.
- ② **Получете** числата a и b .
- ③ Ако $a \neq b$, то изпълнете стъпка ④, иначе – ⑥.
- ④ Ако $a > b$, то изчислете $a - b$ и го помнете като a , иначе изчислете $b - a$ и го помнете като b .
- ⑤ Изпълнете стъпка ③.
- ⑥ **Върнете като резултат** стойността на a .
- ⑦ **Продължете** основната работа.

ПРОГ_03

13/28

ОПИТ ЗА ОПРЕДЕЛЕНИЕ НА АЛГОРИТЪМ

Алгоритъм е точно предписание, което **задава изчислителен процес** (наричан в този случай **алгоритмичен**), **започващ** **от произволни начални данни** (принадлежащи на характерно за алгоритъма **множество от входни данни**) и предназначен за **получаването на напълно определен** от тези начални данни **резултат**.

ПРОГ_03

14/28

ОПИТ ЗА БАЛАНСИРАНО ОПРЕДЕЛЕНИЕ

Алгоритъм е абстрактен модел на обекти и на действия, които могат да се прилагат над тези обекти, за постигане на определена цел (резултат).

Като модел на определен процес, всеки алгоритъм съдържа **два основни момента:**

- ① декларативно описание** (модел на участващите **обекти и отношенията** между тях);
- ② процедурно описание** (на моделирания **процес**).

ПРОГ_03

15/28

КОМПЮТЪРЕН АЛГОРИТЪМ

Всеки алгоритъм в компютърната информатика (**компютърен алгоритъм**) е **множество от недвусмислени и изпълними стъпки** (описание на обекти и действия), **моделиращи краен и ограничен** в пространството и времето **информационен процес**, при който **от дадена начална (входна) информация се получава** като резултат **друга (изходна) информация** — в по-късен период и евентуално на друго място.

ПРОГ_03

16/28

ПРОГРАМА

Програмата представя даден алгоритъм във форма и вид, в които той може да бъде възприет и съответно изпълнен от автомат (компютърна система).

Съставянето на програми е част от процес, наречен програмиране.

ПРОГ_03

17/28

ПАРАМЕТРИ НА ВСЕКИ АЛГОРИТЪМ

- ❶ множество (област) на възможните **начални** (входни) **данни**;
- ❷ множество (област) на възможните **резултати** (изходни) данни;
- ❸ множество на **междинните резултати**;
- ❹ правило за започване (**начало**);
- ❺ правила за непосредствена **обработка** (за получаване на междинните резултати);
- ❻ правило за **край** (условие за приключване на изпълнението на алгоритъма);
- ❼ правило за **посочване на резултата**.

ПРОГ_03

18/28

СВОЙСТВА НА КОМПЮТЪРНИТЕ АЛГОРИТМИ

- ❶ формалност;
- ❷ крайност;
- ❸ дискретност (стъпковост);
- ❹ определеност (детерминираност);
- ❺ масовост;
- ❻ изпълнимост;
- ❼ ефективност;
- ❽ резултатност.

ПРОГ_03

19/28

ФОРМАЛНОСТ

Не е необходимо **изпълнителят** да има представа за решаваната задача и естеството на **получаваните резултати** – **достатъчно е** той да **изпълнява** една след друга **предписаните** му елементарни **операции** (команди).

Това свойството е от **съществено значение**, защото позволява **изпълнителят** на един алгоритъм да **бъде** и лишен от разум **автомат**.

ПРОГ_03

20/28

КРАЙНОСТ

Алгоритъмът трябва да бъде **описание на процес**, който е **локализиран** в пространството **и краен** във времето.

От това свойство следва, че **всяко изпълнение** на алгоритъма **задължително има** освен **начало** още **и край** във времето.

Нарушаването на това изискване при описанията на алгоритми (а от там и при тяхното изпълнение на компютри) **води** например **до проблем**, известен сред специалистите по КИ като „**зацикляне**“.

ПРОГ_03

21/28

ДИСКРЕТНОСТ

Свойството е свързано с обстоятелството, че **описанието**, представено от един алгоритъм, **се състои от краен брой елементи** (декларации, обекти, инструкции, команди и др.), а **съответният алгоритмичен процес** (**изпълнението**) **протича на отделни стъпки**.

Това свойство **налага непрекъснатите** по своята природа **процеси и обекти** да **се моделират чрез дискретни компютърни представления**.

ПРОГ_03

22/28

ОПРЕДЕЛЕНОСТ

Означава, че **на всяка стъпка информацията за състоянието** и протичането на моделирания процес трябва да **е достатъчна, за да определи еднозначно следващото действие** на изпълнителя.

Следствие: Ако процесът е краен, то **резултатът е напълно определен само от началните (входни) данни и действията**, описани от алгоритъма.

ПРОГ_03

23/28

МАСОВОСТ

Отразява възможността при изпълнението на алгоритъма **за всеки начален елемент (от допустимото множество входни данни)** да **се получава търсеният резултат**.

Иначе казано, един **алгоритъм** може да **се прилага не само при** решаването на **една конкретна задача, а на цял клас** от еднотипни **задачи**.

ПРОГ_03

24/28

ИЗПЪЛНИМОСТ

Силно изискване, което се поставя на компютърните алгоритми, е **да се състоят от „изпълними стъпки“**.

Изискването за изпълнимост е **условно** от гледна точка на **знанията и уменията на конкретния изпълнител**.

Границата между „**изпълнимо**“ и „**неизпълнимо**“ обаче е **доста условна** и от друга гледна точка.

ПРОГ_03

25/28

ЕФЕКТИВНОСТ

Алгоритмичният процес е **ефективен**, ако **приключва в „реално“ време** и **всички** (присъщи на алгоритъма) **резултати** се получават **след „приемлив“ брой стъпки**.

Не може да се посочи **коя е „реалната“ граница** между **ефективни** и **неефективни** алгоритми и **в какво се измерва ефективността** по време (в секунди, минути, часове или дни).

A. $2^{32} = 2(\underbrace{2(2(\dots 2)\dots)}_{31 \text{ умножения}})$	срещу	B. $2^{32} = (\underbrace{(((2^2)^2)^2)^2}_{5 \text{ умножения}})^2$
--	-------	---

ПРОГ_03

26/28

РЕЗУЛТАТНОСТ

Завършването на изпълнението на един алгоритъм **е осигурено** (за произволни начални данни от областта на входното множество) **след краен брой операции**.

Резултатността понякога се третира като **насоченост**: **след краен брой стъпки** трябва да се получи **или решението или индикация за неприложимостта** на този алгоритъм към тези входни данни.

ПРОГ_03

27/28

**БЛАГОДАРЯ ВИ
ЗА ВНИМАНИЕТО!**

**БЪДЕТЕ С МЕН И
В СЛЕДВАЩАТА ЛЕКЦИЯ,
КОЯТО ЩЕ НИ ОТВЕДЕ
В НЕВЕРОЯТНИЯ СВЯТ НА
КОМПЮТЪРНИТЕ
СИСТЕМИ**