

Пример 1. Да разгледаме процедурата за сравняване на две естествени числа, представени в позиционна бройна система с основа P . Основните обекти на тази процедура са цифрите на P -ичната бройна система, в която са представени числата. Единствената операция, от която имаме нужда, е сравняването на две P -ични цифри a и b . Резултатът от това сравнение е една от следните три възможности: a е по-малко от b , a и b са равни, a е по-голямо от b . Процедурата действително решава една масова задача, защото е в състояние да сравни произволни числа независимо от тяхната големина или кое да е друго тяхно свойство. Вход за процедурата са две последователности от P -ични цифри, представящи двете числа m и n , определящи дължините (в брой цифри) на всяка от двете последователности. Изход на процедурата ще бъде един от трите възможни резултата от изпълнението на операцията сравняване на цифри.

Ако едно от двете числа е било по-късо от другото, процедурата завършва, като го обявява за по-малко. В противен случай разгледаната процедура, започвайки отляво надясно, сравнява цифрите на двете числа до несъвпадение на цифри или до достигане края на последователностите от цифри. Ако се срещне несъвпадение на цифри, процедурата обявява за по-малко това от двете сравнявани числа, в което е по-малката от двете цифри. Ако всички цифри на двете числа съвпадат, те са равни.

Тъй като единствената прилагана операция е добре дефинирана, изходът е предопределен. Освен това алгоритъмът винаги завършва в не повече от $\max\{m, n\}$ стъпки. Следователно тази процедура е крайна и детерминирана. Цикличният ѝ характер е очевиден. Всичко това е основание да я наречем алгоритъм.

Подобен анализ можем да направим и на останалите споменати по-горе процедури и да видим, че те също имат качествата на алгоритми.

Пример 2. Ето как изглежда (с малки съкращения, за да стане по-лесен за четене) алгоритъмът за събиране на две числа в десетична бройна система, създаден по всяка вероятност от древни индийски математици и достигнал до нас благодарение на трактата на Ал Хорезми: „Ако искаш да прибавиш число към

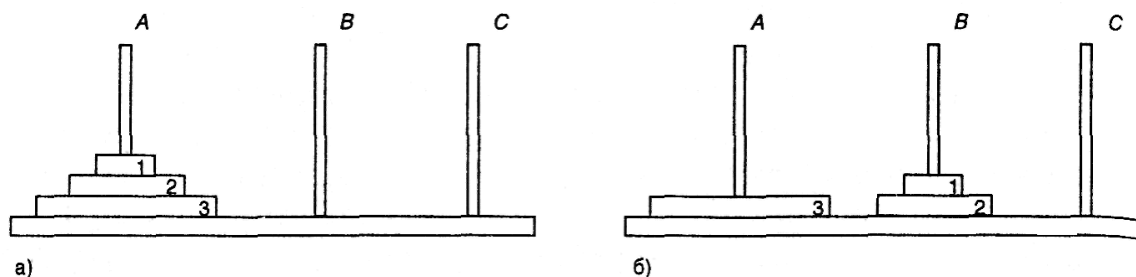
число..., постави двете числа в два реда, т.е. едно под друго и нека бъде разрядът на единиците под разряда на единиците и разрядът на десетиците под разряда на десетиците. Ако искаш да събереш двете числа, то ще събереш всеки разряд със съответния му, който е над него, т.е. единиците с единиците, десетиците с десетиците. Ако в някой от разрядите... се събере десет, тогава постави единица в следващия разряд, например ако имаш в единиците десет, то сложи единица в десетиците и там тя ще означава десет. Ако от числото е останало нещо или самото то е било по-малко от десет, остави го в същия разряд. Ако нищо не остане, постави в разряда кръгче (нула), за да не остане разрядът празен... и да се приеме вторият за първи и да се излъжеш в числото си. Точно така ще направиш с всичките разряди..." (М. ал Хорезми, „Математическите трактати, Фан, Ташкент, 1983).

В това описание се открояват всички характеристики на алгоритъма, които посочихме. Очевидна е цикличната структура на процедурата. Операциите са събиране, изваждане на 10 и сравняване на естествени числа. Броят на стъпките се определя еднозначно от броя на цифрите $\max\{m, n\}$ в по-голямото от двете числа и се изразява с $k_1 \cdot \max\{m, n\} + k_2$, където k_1 и k_2 са константи.

Пример 3. В живота се срещат процедури, които много приличат на алгоритми. Ето част от описанието на една такава процедура, целта на която е да се приготи ястието пържени патладжани: „Посолете нарязаните патладжани, потопете ги в счукал сухар или брашно и ги изпържете в силно сгорещено олио до златисто оцветяване." Такава процедура не може да бъде наречена алгоритъм. Изборът на сухар или брашно е оставен на изпълнителя. Степента на сгорещяване на олиото не е определена точно. Не е фиксиран и краят на процедурата, защото различните изпълнители могат да разбират по различен начин думите „златисто оцветяване".

Пример 4. Да разгледаме следната задача, известна като Ханойски нули. Дадени са три диска, означени с числата 1, 2 и 3, които имат различни радиуси, като радиусът на диска 1 е по-малък от този на 2, а радиусът на 2 - по-малък от този на 3. Дадени са и три пръчки, означени с А В и С. Дисковете са продупчени в

центъра и са нанизани на пръчката А, като най-отдолу е 3, после - 2, а най-отгоре е 1. Допуска се само следната операция: диск може да се премести от една пръчка на друга, ако най-отгоре на втората пръчка е поставен диск с по-голям радиус от този на премествания. Задачата е: само с допустими премествания всички дискове да отидат от пръчката А на пръчката С. Началното положение на дисковете е показано на фигура а):



Ханойски кули

Ще опишем алгоритмична процедура за решаване на задачата. Преместването на диска от върха на пръчката x върху пръчката y да означим с $\text{move}(x, y)$. Да преместим първо най-малките два диска от А на В със следните три стъпки:

$\text{move}(A, C)$

$\text{move}(A, B)$

$\text{move}(C, B)$

Текущото разположение на дисковете е показано на фигура б)

Сега с $\text{move}(A, C)$ преместваем най-големия диск на пръчката С. Остава да преместим двата най-малки диска от В върху С. Това става по начин, подобен на този който преместихме тези два диска от А на В, само че в ролята на А е В, в ролята на В е С, а в ролята на С е А:

$\text{move}(B, A)$

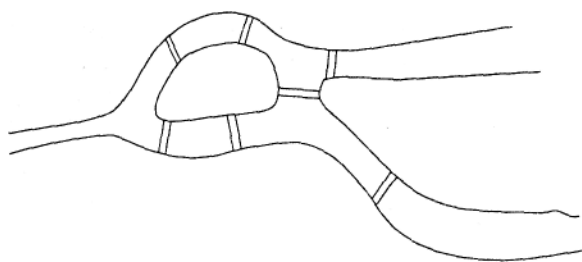
move(B, C)

move(A, C)

Забележете, че мястото, на което се намира най-големият диск, е без значение, защото всеки друг диск може да бъде поставен върху него.

Задачата за Ханойските кули с три диска трудно може да се нарече масова. Тя би станала такава, ако дисковете са не 3, а произволен предварително зададен брой n . За да получим алгоритъм за решаване на масовата задача, нека първо видим как идеята, приложена по-горе, може да се използва, ако на пръчката А са поставени 4, а не 3 диска. Да преместим първите три диска от А на В по начин, подобен на този от алгоритъма по-горе (пръчките В и С разменят ролите си), след това най-големият се премества на С и отново трите най-малки се преместват, сега от В на С. Същата идея може да се приложи и за произволен брой n диска.

Пример 5. Ето още една задача, известна като *Кьонигсбергски мостове*. През 1736 г. знаменитият математик Леонард Ойлер живеел в пруския град Кьонигсберг (днес Калининград в Русия). Реката Прегел разделя града на две части и образува два острова. Седем моста свързват двата бряга с двата острова и островите помежду им, както е показано на следващата фигура:



Мостовите на Кьонигсберг

Леонард Ойлер дълго се опитвал да излезе от дома си, да се разходи из града, като мине по всеки мост точно веднъж и се върне отново в дома си. Всички опити завършили неуспешно. Предизвикан от този неуспех, великият Ойлер успял да докаже математически, че не съществува такъв начин за преминаване по мостовите

на Кьонигсберг и следователно никой алгоритъм не би успял да реши тази задача. В същото време Ойлер показал, че при определена конфигурация на парчетата суша и свързващите ги мостове, съществува алгоритъм за такова, днес наричано Ойлерово обхождане.

Пример 6. Блестящ пример за нечислов алгоритъм ни дава аржентинският писател Хулио Кортасар в краткия си разказ „Указания за изкачване на стълба" (Х. Кортасар. „Междуетажие". Народна култура, София, 1985):

„Едва ли някой е пропуснал да отбележи, че нередко земята се нагъва по такъв начин, щото една част се качва под прав ъгъл спрямо земната плоскост и после следващата част застава успоредно на тая плоскост, за да даде път на нова отвесна - поведение, което се повтаря в спирала или в начупена линия до крайно променливи височини. Прикляквайки и поставяйки лявата ръка върху една от отвесните части, а дясната върху съответната водоравна, вие сте в моментно владение на едно стъпало или степен. Всяко едно от тия стъпала, образувано, както се вижда, от две съставки, е разположено малко по-нагоре и малко по-напред, отколкото предхождащото, ръководно начало, което осмисля стълбата, понеже всяко друго съчетание би произвело форми може би по-красиви и живописни, ала неспособни да пренасят от приземен на първия етаж.

Стълбите се изкачват с лице напред, тъй като заднишком или странишком се оказват особено неудобни. Естественото положение е да се държите на крака, ръцете увиснали без усилие, главата повдигната, макар не толкова, че очите да не виждат стъпалата, непосредствено над онова, върху което стъпвате, и да се диша бавно и равномерно. Изкачването на стълбата се започва, като повдигнете тая част на тялото, разположена вдясно долу, обвита почти винаги в кожа или велур, и която, освен при изключения, се побира точно върху стъпалото. След като е поставена върху първото стъпало въпросната част, която за по-кратко ще наречем ходило, свива се частта, равностойна на лявата (също наричана ходило, но която не трябва да се бърка с гореупоменатото ходило), и издигайки я до височината на ходилото, трябва да продължи, докато бъде поставена върху второто стъпало, при което на последното ще почине ходилото, а на първото ще почине ходилото. (Първите стъпала са винаги най-трудните, докато се придобие необходимата съгласуваност. Съвпадението на име между ходилото и ходилото затруднява обяснението. Внимавайте особено да не повдигате едновременно ходилото и ходилото.)

Щом стигнете по този начин на второто стъпало, достатъчно е да повтаряте последователно движенията, докато се озовете на края на стълбата. От нея се излиза лесно с лек удар на пета, която я закрепва на мястото ѝ, от което не ще мръдне до момента на спускането.”

Всичко, което споменахме по-горе за същността на понятието алгоритъм, е почувствано и отразено в този текст. И необходимостта от точно определяне на обектите и операциите (включително и опасностите от неточни дефиниции), и крайността на процедурата, и цикличният ѝ характер.

[към тема 11](#)