РАЗШИРЕН АЛГОРИТЪМ НА ЕВКЛИД

Кинка Иванова Кирилова-Лупанова, Цеца Илиева Байчева

РЕЗЮМЕ

Съвременната информатика използва съществено вековните постижения на математиката. Редица области на математиката са в основата на информатиката. Ясно е, че между двете науки има неразривна връзка. Интегрирането на елементи от училищния курс по математика в курса по информатика може да се реализира благодарение на обектите, с които оперират двете науки; аналогията в учебното съдържание и др.

Тази статия представя използването на разширения алгоритъм на Евклид при решаване на задачи по информатика при подготовка за състезания и олимпиади.

Ключови думи: разширен алгоритъм на Евклид, информатика, математика.

Тази статия е предназначена за ученици, които са усвоили цикличните конструкции на избран от тях език за програмиране в часовете от профилирана подготовка по информатика, могат да прилагат на практика алгоритъма на Евклид за намиране на най-голям общ делител на две цели числа и се подготвят за участие в състезания и олимпиади по информатика. Целта е да се реализира програмно и да се прилага за решаване на задачи разширеният алгоритъм на Евклид.

ВЪВЕДЕНИЕ

Едни от основните цели на профилираното обучение по информатика в средното училище са свързани със създаване на алгоритмично мислене, изграждане на умения и способности в учениците, които им дават възможност технологично да представят своите идеи. За постигането на поставените цели много често се използват междупредметни връзки с математиката като се решават математически задачи с компютър. Ще отбележим, че когато през осемдесетте години на миналия век се организират първите олимпиади и състезания по информатика предлаганите задачи са били предимно с математическа насоченост. Математическата тематика доминира и в някои от първите публикации посветени на извънкласната работа по информатика. Първи опити в това отношение са направени в

Ръководство за извънкласна работа по информатика на базата на езика БЕЙСИК (Рахнев & Гъров & Гавраилов, 1985), Програмиране на рекурентни формули (Рахнев & Гъров, 1988), Някои задачи по програмиране свързани с числата на Фибоначи (Рахнев & Гъров, 1988) и др. При преподаване на циклични алгоритми в 9 клас се разглежда алгоритъма на Евклид за намиране на най-голям общ делител на две цели числа с изваждане и с деление. Едно естествено продължение на усвояването на този алгоритъм в часовете по СИП е реализация на разширения алгоритъм на Евклид и неговото приложение за решаване на диофантови уравнения от първа степен. Разглеждането на тази тема затвърждава знанията на учениците за цикличните конструкции на езика за програмиране, формира ключови умения в областта на алгоритмичното излагане на разсъжденията, дава възможност за разгръщане на творческия потенциал на учащите се и ги мотивира за изява в състезания и олимпиади по информатика.

ФОРМУЛИРАНЕ НА ЗАДАЧАТА

Целта е, след като е усвоен алгоритъма на Евклид, същият да се разшири така, че не само да намира НОД на две цели числа, но и целите числа ***x*** и ***y***, такива че ***ax+by=НОД(a,b).*** Тези числа ***x*** и ***y*** се наричат коефициенти на Безу.

Идеята на алгоритъма е в това, че на всяка стъпка се съхраняват коефициенти, изразяващи текущите стойности на ***a*** и ***b*** чрез изходните числа ***a*** и ***b.***

Да намерим НОД на числата 120 и 23, като използваме алгоритъма на Евклид с деление.

Делимо	**Делител**	**Частно**	**Остатък**
120	23	5	5
23	5	4	3
5	3	1	2
3	2	1	1
2	1	2	0

В този случай, остатъкът в последния ред (който е 0), показва че НОД(120,23)=1.

Тази таблица може да запишем по следния начин. Да започнем изразяването на остатъците от таблицата.

Първият ред е същият, като на оригиналното уравнение с членове 120 и 23. Обаче от втория ред на надолу, стойностите на текущото делимо и текущия делител намаляват. При това се забелязва, че:

Остатък	=	**Делимо**	-	**Частно**	×	**Делител**
5	=	120	-	5	×	23
3	=	23	-	4	×	5
2	=	5	-	1	×	3
1	=	3	-	1	×	2
0	=	2	-	2	×	1

- На всеки ред делителят е остатък от предходния ред

- Делимото е делител от предходния ред

По такъв начин членовете могат да бъдат заместени с предходните два реда. Следователно всеки остатък може да се представи многократно като сума на двете първоначални числа (в случая 120 и 23).

Съответно делителят на третия ред е остатък от втори ред. По-нататък, делимото от трети ред - /5/, е делител във втория ред и остатък от първи ред.

Следвайки тези разсъждения, остатъкът на всяка стъпка се записва чрез първоначалните стойности 120 и 23.

q			***a*** ***r*** ***b***
5 = 120 - 5 × 23 =			= 1 × 120 - 5 × 23
3 = 23 - 4 × 5	= 1×23	- 4 × (1×120 - 5×23)	= -4 × 120 + 21 × 23
2 = 5 - 1 × 3	= (1×120 - 5×23)	- 1 × (-4×120 + 21×23)	= 5 × 120 - 26 × 23
1 = 3 - 1 × 2	= (-4×120 + 21×23)	- 1 × (5×120 - 26×23)	= -9 × 120 + 47 × 23

1) Първият ред е в желаната форма.
2) Във втория ред заместваме 5 с представянето му от първия ред
3) В третия ред заместваме 5 и 3 с представянията им от първи и втори ред.
4) В четвъртия заместваме 3 и 2 с представянията им от втори и трети ред.

В последния ред се чете $1 = -9\times120 + 47\times23$, което е търсеното решение: $x = -9$ и $y = 47$.

И така намерихме цели числа ***x*** и ***y*** , такива че ***ax+by=НОД(a,b).*** При това на всяка стъпка от алгоритъма се съхраняват коефициенти, изразяващи текущите стойности на ***a*** и ***b*** чрез изходните числа ***a*** и ***b.*** При замяна на двойката ***(a,b)*** с ***(b,r)*** тези коефициенти се преизчисляват.

ПРОГРАМНА РЕАЛИЗАЦИЯ НА ЕЗИКА С++

Може да представим числата ***a*** и ***b*** по следния начин:

a=1*a+0*b и ***b=0*a+1*b***

Следователно: ако ***x1=1*** и ***y1=0,*** то ***a=a*x1 + b*y1***

и ако ***x2=0*** и ***y2=1***, то ***b=a*x2 + b*y2***

Условието за край на цикъла е както при обикновения алгоритъм на Евклид с деление, когато ***b*** стане ***0***.

Изразяваме остатъка във вид на линейна комбинация на ***a*** и ***b***:

Знаем, че ***r=a-q* b =a.x1+b.y1-q.a.x2-q.b.y2=a.(x1-q.x2) +b.(y1-q.y2)⇒***

x2=x1-q.x2 и ***y2=y1-q.y2*** ***x2*** ***y2***

Ако смятаме по таблицата и по алгоритъма, последователно получаваме:

x2=1, y2=-5
-4 21
5 -26
-9 47

По този начин пресмятаме коефициентите за всеки следващ ред. При завършване на цикъла търсените числа са: НОД=a, x1 и y1.

Предлагам и самата програма:

```
void extended_euclid(int a, int b, int &x1, int &y1, int &d)
{  int q, r, x2,  y2,  t;
   x1 = 1; y1 = 0;  x2 = 0; y2 = 1;
  while (b != 0)
  {q = a/ b;  r = a % b; a=b;b=r;//замяна на двойката (a,b) с (b,r)
 //изчисляване на новите стойности на x1 и x2
    t=x2; x2=x1-q*x2; x1=t;
    // изчисляване на новите стойности на y1 и y2
    t=y2; y2=y1-q*y2; y1=t;
    }  d=a; }
int main()
{int a,b,d,x,y; cin>>a>>b;
extended_euclid( a,  b, x, y, d); cout<<"x="<<x<<"y="<<y<<"d="<<d;
return 0; }
```

ПРИЛОЖЕНИЕ НА РАЗШИРЕНИЯ АЛГОРИТЪМ НА ЕВКЛИД ЗА РЕШАВАНЕ НА ДИОФАНТОВИ УРАВНЕНИЯ

Знаем, че намирането на НОД на две цели числа се използва при решаването на уравнението $ax+by=c$ с цели коефициенти в множеството на целите числа. Това уравнение се нарича **диофантово** уравнение от първа степен с две неизвестни.

Необходимото и достатъчно условие да е решимо, е НОД(a,b) да дели c. В този случай уравнението има безброй много целочислени решения.

Едно решение (x_0, y_0) може да намерим и като използваме разширения алгоритъм на Евклид. Тогава други решения се получават чрез :

$x=x_0+bk$ и $y= y_0 – ak$, където k е произволно цяло число.

Проблемът е как ще намерим x_0, y_0 ?

1.) Намираме d = НОД(a,b).

2.) Ако НОД$(a,b)=c=0$, то произволни x и y са решения

3.) Ако d не дели c, то уравнението е нерешимо.

4.) Ако d/c, съкращаваме коефициентите в уравнението: $a_1=a/d$, $b_1=b/d$, $c_1=c/d$ и получаваме уравнение с взаимно прости a, b, c.

5.) Намираме x и y, такива че $ax+by=1$

6.) Умножаваме x и y на c и намираме частно решение x_0, y_0.

7.) Общото решение тогава има вида: $x = x_0 +k*b$, $y = y_0 –k*a$.

```
int main()
{int a,b,d,x,y,c,i,x0,y0;
 cin>>a>>b>>c;
 d=nod(a,b);
```

```
if (d==0) cout<<"произволни x и y са решения";
else  {if (c%d==0) //има решение
            {extended_euclid(a,b, x, y);
              x0=x*c;y0=y*c;
             //други решения
             for (i=-c;i<=c;i++)
             {x=x0-b*i;
              y=y0+a*i;
              cout<<x<<"   "<<y<<endl;
             }     }
else cout<<"няма решение\n"; }
return 0;
}
```

ЗАДАЧИ ЗА САМОСТОЯТЕЛНА РАБОТА

Задача 1. Да се реализира разширения алгоритъм на Евклид като се използва алгоритъма на Евклид за намиране на най-голям общ делител на две цели числа с изваждане.

Задача 2. БАНКА

Една банка разполага с неограничен брой банкноти от *n* различни номинални стойности (например: 2, 3, 5 лв. и т. н.). Напишете програма **banka**, която намира минималния брой банкноти от най-много два различни номинала, с които може да бъде изплатена сума от *s* лева, самите номинали и съответния брой банкноти.

Вход	**Изход**
3	13
3 5 10	3 6
88	10 7

Задача 3. ВАРЕЛИ

Разполагаме с 3 варела с целочислени обеми a, b, c литри, празен варел и кран с вода. Може ли с помощта на тези варели да налеете в празния варел s литра вода? Отпечатайте възможните начини.

Вход	**Изход**
6 2 3 9	1 0 1
	0 3 1
	0 0 3

Задача 4. ВАРЕЛ

При едно от наводненията миналата година реката отнесла всички прибори от къщата на Тошко. Останали само две ведра с различна вместимост и един варел. С ведрата може да се добавя или изчерпва вода. Всеки ден се налагало да се налее във варела определено количество вода.

Тази работа била възложена на Тошко и той изпразвал варела всеки ден и започвал да налива вода с едното или с двете ведра, да изчерпва с едното или с двете, и т.н. докато във варела се налее необходимото количество вода. Вие можете да помогнете на Тошко да не прави излишни движения, като напишете програма **barrel,** която да отпечатва последователността от действия за отмерване на **s** литра вода чрез двете ведра, като общото количество на прехвърлената вода да бъде минимално и извършените операции да бъдат най-малко. Едното ведро има вместимост **a** литра, а другото **b** литра. Ако не е възможно да се отлее исканото количество вода, програмата да съобщава, че няма решение.

Вход	**Изход**
3 5	3
1	11
	+1
	+1
	–2

РЕЗУЛТАТИ, ИЗВОДИ И ПРЕПОРЪКИ

Програмната реализация е на езика C++. Но за да се овладее език за програмиране не е достатъчно да се изучават неговите елементи сами по себе си. Необходимо е да се усвояват основните похвати на програмиране и известни алгоритми за решаване на програмистки задачи. Смятам, че описаната реализация и посочените задачи имат универсален характер и могат да бъдат полезни при програмиране на други езици. От друга страна, разширеният алгоритъм на Евклид провокира алгоритмичното мислене на учениците, развива техните програмистки умения. След усвояването му те, без затруднения, ще могат да го прилагат при решаване на по-сложни и разнообразни задачи.

Изложеният материал е преподаван в продължение на шест години в рамките на две или три занятия от по два учебни часа в СИП за 8-и или 9-и клас. Тази тема беше разгледана на две Национални лагер-школи по информатика в гр. Бургас през 2007 и 2008 г. пред ученици от 7-и клас. Придобитите знания се прилагат успешно при решаване на конкретни практически задачи, както и по време на състезания и олимпиади.

ЛИТЕРАТУРА

http://en.wikipedia.org/wiki/Extended_Euclidean_algorithm

http://alglib.sources.ru/diophant/diophant.php

Рахнев А., Гъров К., Гавраилов О., Ръководство за извънкласна работа по информатика на базата на езика БЕЙСИК, Издателство на МНП, София, 1985.

Рахнев А., Гъров К., Програмиране на рекурентни формули, сп. Математика, бр. 4, 1988.

Рахнев А., Гъров К., Някои задачи по програмиране свързани с числата на Фибоначи, сп. Математика, бр. 8, 1988

Grozdev, S. For High Achievements in Mathematics. ARO, Sofia, 2007.

Учебна програма по информатика – IX клас, профилирина подготовка, МОН, 2003

Кирилова-Лупанова К., „Едно приложение на алгоритъма на Евклид", Математика и информатика, 2008, брой 3

Кинка Иванова Кирилова-Лупанова,
ПМГ „Васил Друмев"- Велико Търново, kkicak@gmail.com
Цеца Илиева Байчева,
ПМГ „Васил Друмев"- Велико Търново, zezab1@gmail.com

THE EXTENDED EUCLIDEAN ALGORITHM

Kinka Ivanova Kirilova-Lupanova, Ceca Ilieva Baycheva

ABSTRACT

Modern informatics substantially use century-old achievements of mathematics. Many parts of mathematics are in the base of informatics. It is clear that there is a strong connection between both sciences. Integration of elements from the school course of mathematics in the course of informatics can be realized due to objects by which both sciences operate, by analogy in the educational content, etc.

This paper presents the use of the extended Euclidean algoriths for solving informatics problems in preparation for competitions and Olympiads.

Kinka Ivanova Kirilova-Lupanova,
High School of Mathematics and Natural Sciences "Vasil Drumev"–
Veliko Tarnovo, kkicak@gmail.com
Tsetsa Ilieva Baycheva,
High School of Mathematics and Natural Sciences "Vasil Drumev"–
Veliko Tarnovo, zezab1@gmail.com