АЛГОРИТМИЧНОТО ЛИЦЕ НА ДРОБИТЕ

Бойко Банчев

Институт по математика и информатика – БАН
boykobb@gmail.com

Резюме. *Съществуват интересни, прости, но недобре познати отношения, структури и алгоритми над дроби. Въвеждането им в обучението по математика и информатика може да обогати и двете дисциплини. За илюстрация на това разглеждам постановката и решаването на някои характерни задачи върху дроби с алгоритмичен характер.*

Ключови думи: *аритметика, дроби, алгоритми, обучение*

Дробите – числови и алгоритмични обекти

Аритметиката на целите и дробни числа заслужено заема устойчиво място в училищното образование по математика. Тя не само предоставя практически полезни знания, но и е сред най-достъпните за всеки ум и възраст математически дисциплини. Г. Харди посочва [3], че:

> „*Елементарната теория на числата е един от най-добрите за начално обучение по математика предмети. Необходимите за нея предварителни знания са малко, предметната ѝ област е ясна и позната. Схемите за построяване на доказателства са прости, общи и малко на брой. А сред математическите науки тя в най-голяма степен разчита на естественото човешко любопитство.*“

Алгоритмичната страна на аритметиката също не е непозната, но е по-малко застъпена в заниманията по математика. Особено подчертано това се отнася до алгоритмите с дроби, които почти напълно отсъстват. Алгоритмичната аритметика е и тази част на традиционната математика, която е най-естествен източник на теми и задачи за информатиката и в частност програмирането: представянето и действията с числа са налице във всеки език за програмиране. С оглед на това естествено е да се потърси по-широко присъствие на този материал в заниманията и по математика, и по информатика в училище.

Да разгледаме следния пример, може би най-непосредствено възникващата задача за аритметична дроб: привеждането в несъкратим вид. За него е необходимо намиране на най-голям общ делител, за което пък можем да се възползваме от знаменития алгоритъм на Евклид или от т.нар. „двоичен алгоритъм“ (в който съществена роля играе деленето на 2). Тук се спирам на първия, тъй като негова разширена форма в действителност дава възможност да се решат няколко задачи наведнъж.

Следващият текст е на езика C и с негова помощ по зададени цели *a* и *b* се пресмятат и отпечатват нод(*a*,*b*), нок(*a*,*b*), несъкратимата форма на *a*/*b*, елементите *q* и приближенията c_2/c_1 от представянето на *a*/*b* като верижна дроб $[q_0,q_1,q_2,\ldots] = q_0+1/(q_1+1/(q_2+1/(\ldots)))$, както и общото решение (*x*,*y*) на целочисленото линейно уравнение *ax*–*by*=нод(*a*,*b*).

```
for (
c1o = 1, c1 = 0
,c2o = 0, c2 = 1
,e = 0
;
q = a/b, t = a%b, a = b, b = t
,t = c1o+q*c1, c1o = c1, c1 = t
,t = c2o+q*c2, c2o = c2, c2 = t
,printf("q=%d, f=%d/%d\n",q,c2,c1)
,b
;
e = !e
) ;
if (!e) c1o = -c1o, c2o = -c2o;
printf("%d/%d=%d/%d\n",c2*a,c1*a,c2,c1);
printf("нод(%d,%d)=%d\n",c2*a,c1*a,a);
printf("нок(%d,%d)=%d\n",c2*a,c1*a,c1*c2*a);
printf("(%d+%dk)%d-(%d+%dk)%d=%d"
,c1o,c1*a,c2*a,c2o,c2*a,c1*a,a);
```

Например за *a*=18, *b*=30: нод=6, нок=90, верижната дроб *a*/*b* = [0;1,1,2]= 0+1/(1+1/(1+1/2)), приближенията са 0/1=0, 1/1=0+1/1, 1/2=0+1/(1+1/1) и (точната стойност) 3/5 – всяко в несъкратим вид. Решението на уравнението *18x–30y*=6 е *x*=2+30*k*, *y*=1+18*k* и се дава от предпоследните стойности 2 и 1 на c_1 и c_2: знаменателя и числителя на предпоследното приближение.

Такова разширяване на Евклидовия алгоритъм лесно се извежда след запознаване с представянето на дроб като верижна и със свойствата му. Алгоритъмът на Евклид се използва и за други пресмятания, а освен това служи и чисто математически като инструмент за доказване на различни твърдения за числа – отличен пример за дълбоките концептуални корени на математиката в алгоритмиката и за практическото преплитане на дейностите в двете дисциплини.

През XIX в. биват открити някои структури, образувани от множества от дроби. Чрез тях – най-вече чрез т.нар. редици на Феъри (Фарей) – стават известни редица важни свойства и отношения между обикновени дроби, удивително как незабелязани дотогава. Тези свойства, отношения и структури съчетават чисто аритметично с комбинаторно съдържание, давайки възможност за едно доста ново, в голяма степен алгоритмично, третиране на множеството от рационалните числа.

Представени в контекста на една или друга структура обаче, тези свойства и отношения се оказват хаотично, непълно и в голяма степен изкуствено въведени, докато те всъщност са присъщи на дробите независимо от всяка конкретна структура. В [1] обърнах внимание именно върху тази аномалия, като заедно с това предложих концептуално изчистване, подреждане и допълване на съответните понятия, факти и връзки между тях. Така организирана, материята добива относителна завършеност и е напълно достъпна за преподаване в която и да е училищна форма.

На основата на тази предварително свършена работа по математическия аспект на третирания материал, като нейно естествено продължение могат удобно да се формулират и решават задачи, вече с предимно алгоритмичен характер, като разглежданите по-долу.

В следващия раздел излагам в резюме резултатите от [1]. Въведените понятия и отношения дават възможност да се формулират и решават редица задачи, някои от които са кратко разгледани по-нататък.

Понятия и свойства, свързани с дроби

Основополагащо значение имат понятията *разстояние* между дроби, *съседство* между дроби и *медианта*, определени както следва.

За дроби (в несъкратим вид) $p/q < r/s$ разглеждаме разстояние между тях:

$$|p/q, r/s| = qr - ps \geq 1$$

и когато $|p/q, r/s| = 1$ наричаме дробите съседни и пишем $p/q \perp r/s$.

Медианта на две дроби $p/q < r/s$ наричаме дробта

$$m = (p+r)/(q+s)$$

– това е единствената дроб, чиито разстояния до p/q и r/s са равни помежду си; тогава те са равни и на $|p/q, r/s|$. Освен това $p/q < m < r/s$. Когато $p/q \perp r/s$, m е несъкратима дроб, съседна е и с p/q, и с r/s и е единствената такава дроб.

С изключение на $p/q = 0/1$, $r/s = 1/0$, налице е или $p \leq r$ и $q \leq s$, или $r \leq p$ и $s \leq q$. Тогава можем да образуваме и *медиразлика*, съответно или $(r-p)/(s-q) > r/s$, или $(p-r)/(q-s) < p/q$. Медиразликата също е съседна с образуващите я дроби.

Следователно интервал от съседни дроби може да се:

– подразделя чрез намиране на все нови и нови медианти; всяка дроб в интервала се получава като медианта на някакво равнище на подразделянето му;
– разширява наляво или надясно чрез намиране на медиразлики.

Очевидно всички дроби в интервал между съседни дроби имат числители и знаменатели, по-големи от тези на всяка от двете образуващи ги дроби.

Всички неотрицателни дроби се получават като медианти при подразделяне, започвайки от интервала [0/1,1/0] ≡ [0,∞). Така можем да получим и кой да е интервал с граници съседни дроби. Около всяка дроб можем да намерим произволно близо разположени дроби, съседни помежду си и с дадената.

Обратно, кой да е интервал може чрез намиране на медиразлики да бъде разширен до [0/1,1/0].

Всяка положителна дроб е медианта на точно една двойка неотрицателни съседни дроби. Едната от дробите-родители е родител и на другия родител на същата дроб. Според това дали левият или десният ѝ родител е такъв, всяка дроб може да бъде определена като „лява" или „дясна".

Родителите на дроб, която е цяло положително число, са по-малкото цяло число и 1/0. Във всички други случаи дробта и родителите ѝ принадлежат на един и същ затворен интервал между две цели числа.

За всяка дроб $a/b \in [p/q,r/s]$, ако $d = |p/q,r/s|$, $m = |p/q,a/b|$ и $n = |a/b,r/s|$, то $da = np+mr$ и $db = nq+ms$ и в частност $a/b = (np+mr)/(nq+ms)$.

Двойката разстояния (m,n) взаимноеднозначно съответства с a/b и може да се тълкува като координати на a/b спрямо $|p/q,r/s|$.

Ако $p/q \perp r/s$, то нод(m,n) = нод$(np+mr,nq+ms) = 1$, което означава, че представянето на a/b по-горе е несъкратима дроб, а m/n може да се разглежда като „координатна дроб" на a/b спрямо интервала.

Координатната дроб на медиантата на дробите $f_1,f_2 \in [p/q,r/s]$ е медианта (в несъкратим вид) на координатните дроби на f_1 и f_2.

Разстоянието между дроби може да се изрази чрез координатите им в даден интервал: ако $a/b = (n_1p+m_1r)/(n_1q+m_1s)$ и $c/d = (n_2p+m_2r)/(n_2q+m_2s)$, то $|a/b,c/d| = |m_1/n_1,m_2/n_2|\cdot|p/q,r/s|$. По-специално, ако p/q и r/s са съседни, в сила е $|a/b,c/d| = |m_1/n_1,m_2/n_2|$ – разстоянието между дроби е равно на това между координатните им дроби.

Задачи, свързани с дроби. Структури от дроби

Популярният „Проект Ойлер" [4] предлага множество математикоалгоритмични задачи в широк диапазон на трудност. Задача 73 изисква да се намери броят на дробите между 1/3 и 1/2, чиито знаменатели не надминават 12000. Как може да бъде направено това?

Можем да образуваме всички дроби със знаменатели ≤12000, подбирайки сред тях тези, които попадат в (1/3,1/2). Този груб подход обаче изисква твърде много ресурси.

Забелязвайки, че $1/3 \perp 1/2$ съобразяваме, че всички дроби между тях се образуват чрез медиантно подразделяне на интервала [1/3,1/2] и следователно можем да подразделяме всеки интервал докато получим знаменател >12000. И този метод е ресурсоемък, още повече, че е съществено рекурсивен. Можем обаче да намираме дробите последователно: първо тази най-близо до 1/3, после следващата и така докато стигнем 1/2.

Първата, да я наречем x/y, трябва да има спрямо 1/3 и 1/2 дроб-координата $1/k$, т.е. е от вида $(k+1)/(3k+2)$, при това $3k+2 \leq 12000$. За да бъде x/y най-малката такава дроб, k трябва да е възможно най-голямо, т.е. $k = \lfloor 11998/3 \rfloor = 3999$, или $x/y = 4000/11999$. Понеже x/y е медианта на 1/3 и друга дроб a/b, ясно е, че a/b е медиразликата $(x–1)/(y–3) = 3999/11996$. Сега от съседните x/y и a/b можем да намерим следващата дроб x'/y', както x/y от 1/3 и 1/2 и т.н. – получаваме доста по-удобен от предишните алгоритъм. (Съществува по-добър и от него, при който преброяваме дробите без да ги образуваме, но той изисква други аритметични знания и не се обсъжда тук.)

Ясно е, че по подобен начин можем да намираме и последователно намаляващи дроби. Могат да се формулират и други задачи в този дух, както и задачи за верижни дроби. В споменатия „Проект Ойлер“ например подобни задачи са още тези с номера 33, 64, 65, 71, 72 и 175, наред с други.

Практически полезна е задачата за намиране на най-добро приближение на дадено число (дроб или не) с дроб, чийто знаменател не надминава посочена големина. Както горната, тази задача може да се реши с „груб метод“, но предизвикателство поставя откриването на по-бърз – и именно в това често се проявява спецификата в решаването на задачи в информатиката сравнена с математиката.

За бързо намиране на подходяща стойност тук можем да изберем някаква дроб, която е близка до приближаваната стойност (или е самата тя), но не отговаря на условието да има достатъчно малък знаменател. Дроб с такъв знаменател можем да намерим сред последователните приближения на верижната дроб, съответна на изходната, намирани чрез разширения алгоритъм на Евклид: известно е, че всяко такова приближение е най-близо до изходната дроб сред всички дроби, чийто знаменател е не по-голям от неговия. Така обаче не винаги попадаме на точно нужния резултат.

Например дробта 21/37 има верижно представяне [0;1,1,3,5] и съответните приближения са 0/1, 1/1, 1/2 и 4/7 (без последното, 21/37, което е самата дроб): можем да изберем за най-добро последното от списъка, на което знаменателят не надминава посочената граница. Ако, да речем, тя е 5, избираме 1/2, а ако е

32, избираме 4/7. В първия случай обаче съществува по-добър избор в лицето на 3/5, а в последния – 13/23 и още по-доброто 17/30.

Можем да търсим резултата и с последователно медиантно делене на избран, съдържащ приближаваната стойност интервал, избирайки всеки път подходящия подинтервал: за 21/37, започвайки от [0/1,1/1], намираме [1/2,1/1], [1/2,2/3], [1/2,3/5], [1/2,4/7], [5/9,4/7], [9/16,4/7] [13/23,4/7] и [17/30,4/7]. Така бихме открили 3/5 на третата стъпка, 13/23 – на 7-та, а 17/30 – на 8-та. Ясно е, че така намирането на най-доброто според ограничението приближение е сигурно, но това става с (понякога много) повече пресмятания.

Изкушаващо е да се съчетаят двата посочени метода, на верижните дроби и на медиантното делене, но доколкото ми е известно, не е предложен задоволителен начин за това.

Други задачи, които биха имали добро приложение при алгоритмични занимания с дроби, са намиране на родителите на дадена дроб, определяне на вида (лява или дясна) на дроб и посочване на нейния старши/младши родител. Също интересна възможност е по-подробното изследване на съответствието между дроби и координатните им представяния.

Добра възможност е изследването на дробни структури като редиците на Феъри и няколкото познати дървета от дроби. Във връзка с такива структури естествено възникват задачи за определяне на мястото на дроб или обратно – намиране на дроб по зададено място, а също за откриване и извличане на подструктури и др.под.

Например редица на Феъри от ред *n* е редица от подредените по големина дроби в интервала [0,1], на които знаменателите не надминават *n*. Полезно математикоалгоритмично упражнение е съставянето на ефективна програма за намиране на дроб на определено място в зададена с номера си редица. За целта трябва да се опознаят (за което е добре да се изхожда от приведените в предишния раздел свойства) и използват особеностите на редиците.

Още по-богато на интересни свойства е дървото на Щерн-Броко, което съдържа всички дроби по веднъж. То възниква от описания по-горе процес на подразделяне чрез медиани, започвайки от интервала [0/1,1/0]: първата получена дроб е 1/1, а в общия случай на колкото по-късна стъпка се получава дадена дроб, на толкова по-нисък ред в дървото е тя. Понеже процесът на подразделяне се разклонява двоично, естествено се получава двоично дърво. Във всеки ред дробите са подредени по големина.

Друго двоично дърво, съдържащо всички дроби, е това на Калкин и Уилф, което по-вярно може да бъде наречено „древногръцко дърво на отношенията" [1]. При него левият и десният наследници на коя да е дроб p/q са съответно $p/(p+q)$ и $(p+q)/q$. Познато е и дървото на Шен и Андреев – също двоично и съдържащо всички дроби – където наследници на дроб *p*/*q* са съответно (*p*+*q*)/*q* и *q* /(*p*+*q*).

Всяко от тези дървета има редица интересни свойства, които могат да бъдат предмет на алгоритмични изследвания. Те имат и едно обединяващо свойство. Макар открити в различни времена и с различни мотиви, трите в известен смисъл са еднакви: множеството от числа в кой да е избран ред е едно и също за всяко дърво и съдържа заедно с всяка дроб и реципрочната ѝ.

```
                                1/1

                1/2                             2/1

        1/3             2/3             3/2             3/1

    1/4     2/5     3/5     3/4     4/3     5/3     5/2     4/1

1/5 2/7 3/8 3/7 4/7 5/8 5/7 4/5 5/4 7/5 8/5 7/4 7/3 8/3 7/2 5/1

...............................................................
```

Дърво на Щерн-Броко

```
                                1/1

                1/2                             2/1

        1/3             3/2             2/3             3/1

    1/4     4/3     3/5     5/2     2/5     5/3     3/4     4/1

1/5 5/4 4/7 7/3 3/8 8/5 5/7 7/2 2/7 7/5 5/8 8/3 3/7 7/4 4/5 5/1

...............................................................
```

Дърво на Калкин-Уилф

```
                                1/1

                2/1                             1/2

        3/1             1/3             3/2             2/3

    4/1     1/4     4/3     3/4     5/2     2/5     5/3     3/5

5/1 1/5 5/4 4/5 7/3 3/7 7/4 4/7 7/2 2/7 7/5 5/7 8/3 3/8 8/5 5/8

...............................................................
```

Дърво на Шен-Андреев

Забелязвайки, че всяка по-голяма от 1 дроб може да се получи като реципрочна на дроб, по-малка от 1, може да се сметне за достатъчно, от гледна точка на образуване на всички дроби, да се разглеждат само по-малките от 1. Всяко от горните три дървета, ако в него оставим само такива дроби, поражда друго. Така от дървото на Щерн и Броко остава лявото му поддърво, а от това на Шен и Андреев – дървото, наречено на Кеплер. От дървото на Калкин и Уилф се получава дърво, което няма известно наименование.

```
                        1/2

          1/3                         2/3

    1/4         3/4         2/5             3/5

1/5    4/5    3/7    4/7    2/7    5/7    3/8    5/8

. . . . . . . . . . . . . . . . . . . . . . . . . . . . .
```

Дърво на Кеплер

Една от интересните, но трудни задачи, свързани с дробни дървета, е намирането на ефективни алгоритми за изброяване, т.е. построяване в редица, на дробите в тях. Най-добрият резултат в това отношение е намерената формула $x' = 1/(\lfloor x \rfloor + 1 - \{x\})$, която за коя да е дроб x намира следваща x' и така безповторно обхожда всички дроби; формулата реализира изброяване по редове на дървото на Калкин и Уилф.

Интересно е да се отбележи и геометричното тълкуване на дробите, което създава благоприятни възможности за решаване и на други видове математикоалгоритмични задачи. Ако на всяка дроб съпоставим точка в равнината с координати числителя и знаменателя на дробта, получаваме еднозначнообратимо съответствие между дроби и точки с целочислени координати. Тогава например триъгълникът с върхове (0,0) и кои да е две целочислени точки има лице, равно на половината от разстоянието между съответните две дроби; ако дробите са съседни – 1/2. Специфичен смисъл придобива и образуването на медианта: при него от даден триъгълник се получават два, всеки с лице, равно на това на изходния и т.н.

Обобщавайки възможните посоки за занимания с дроби по информатика, освен алгоритмизиране и програмиране в пряк смисъл, за каквито предимно стана дума дотук, могат да се споменат и други възможности. Една от тях е изобретателното визуализиране на дробни структури, не само на споменатите, но и на такива, които се откриват в процеса на изследване. Интересен пример в това отношение е [2], където редиците на Феъри служат за пораждане на текстури като форма на изкуство. Визуализирането може да служи и на изследователски цели като извършване на наблюдения, търсене на математически зависимости и проверка на хипотези.

В подобен дух могат да бъдат изследвани и връзки между дробни структури и целочислени редици. За последното, освен извършване на математически анализ и съставяне на програми, отличен инструмент е Енциклопедията на целочислените редици [5].

Програмното представяне на структури като споменатите само по себе си също може да се окаже предизвикателство или съдържателна задача. Дори смятането с големи стойности, ако не се поддържа непосредствено от езика за програмиране, изисква допълнителна работа. От друга страна, идеята за представяне на безкрайни структури, каквито са например дробните дървета, може да се използва като мотивиращ пример за запознаване под една или друга форма с лениви пресмятания в програмирането, особено с привличане за целта на подходящ език за програмиране.

Литература

1. B. Bantchev. *Fraction space revisited.* Mathematics and Education in Mathematics, 2012. Proc. 41st Spring Conf. UBM, April 2012, pp. 209-218
2. R. Griswold. *Designing with Farey fractions*, 2001. http://www.cs.arizona.edu/patterns/weaving/webdocs/gre_fry.pdf
3. G. H. Hardy. *An introduction to the theory of numbers.* Bull. Amer. Math. Soc. 35 (1929), 778-818
4. Project Euler. http://projecteuler.net
5. *The on-line encyclopedia of integer sequences.* http://oeis.org

THE ALGORITHMIC FACE OF FRACTIONS

Boyko Bantchev

***Abstract**. A number of interesting and simple but not well known relations, structures and algorithms on fractions exist. Introducing them to the disciplines of mathematics and informatics as taught in school can be advantageous to both these disciplines. To illustrate this, a number of exemplary algorithmic problems on fractions and their solutions are discussed.*