https://doi.org/10.53656/math2023-4-5-rec

РЕКУРЕНТНИТЕ РЕДИЦИ В УЧИЛИЩНИЯ КУРС ПО МАТЕМАТИКА И МЕЖДУПРЕДМЕТНИТЕ ИМ ВРЪЗКИ С ИНФОРМАТИКА И ИНФОРМАЦИОННИ ТЕХНОЛОГИИ

Гл. ас. д-р Бориславa Кирилова, доц. д-р Филип Петров
Софийски университет „Св. Климент Охридски“

Резюме. В статията е разгледан стандартният ход за изучаване на рекурентни редици в училищния курс по математика. Предложено е разширяване на темата в профилирана подготовка чрез въвеждане на изучаване на рекурентни линейни хомогенни редици. Представена е поредица от задачи по информатика и информационни технологии, с които се изграждат двупосочни междупредметни връзки, целящи както да покажат приложната страна на рекурентните редици, така и да мотивират по-изявените ученици за научаване на някои по-специфични знания от основите на информатиката и да се докоснат до някои техники за програмиране.

Ключови думи: рекурентни редици; рекурентно отношение; характеристично уравнение; итерация; рекурсия; математическа индукция; мемоизация; рекурсивна функция

1. Увод

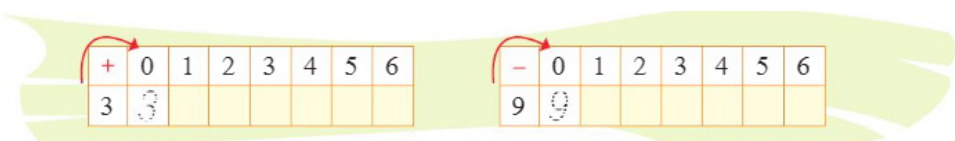
Дефиниция 1. Числовата редица $\{a_0, a_1, a_2, a_3, \dots, a_n, \dots\}, n \in \mathbb{N}$, *$n$-тият член на която може да се изрази чрез предходните му k члена, т.е. съществува такава k-аргументна функция f, че $a_n = f(a_{n-1}, a_{n-2}, \dots, a_{n-k})$, където $n \geq k$, се нарича* ***рекурентна редица*** *от ред $k \in \mathbb{N}, k \geq 1$, а равенството – $a_n = f(a_{n-1}, a_{n-2}, \dots, a_{n-k})$ – **рекурентно отношение** (или **рекурентна зависимост**). Редицата е определена еднозначно, ако са зададени нейните първи k члена, които се наричат начални условия.*

За първи път рекурентни редици се появяват в явен вид в учебните програми на българското училище в X клас (общообразователна подготовка), основно под формата на аритметична прогресия и геометрична прогресия. В профилирана подготовка се разглеждат и други рекурентни редици, но с относително малък хорариум. Основната цел при задачите по темата е да се изрази общият член на редицата $\{a_n\}$ като функция на числото n, която е зададена в явен вид, например чрез алгебричен израз. По този начин изчислително тежката итерация или рекур-

сия се свежда до пресмятане на единичен израз. В настоящата статия ще бъде разгледан стандартният ход за изучаване на рекурентни редици в училищния курс по математика, ще бъдат обсъдени някои възможности за междупредметни връзки с предметите информатика и информационни технологии и ще бъдат предложени възможности за допълване на училищния курс чрез изучаване на рекурентни линейни хомогенни редици в профилирана подготовка.

2. Пропедевтика

Изучаването на рекурентни редици започва в неявен вид в началното училище. Още в I клас децата периодично решават задачи, в които трябва да намерят елементарна зависимост между дадени няколко члена на редица и да пресметнат следващи. Примери за такива задачи от (Garcheva & Manova 2016) са показани на фиг. 1. В първия пример се подразбира редицата $a_n = a_{n-1} + 1$ с първи член $a_0 = 3$, а във втория – редицата $a_n = a_{n-1} - 1$ с първи член $a_0 = 9$. Показани са в явен вид индексите на съответните членове и споменатите зависимости. Липсата на пояснително текстово условие на задачата не се очаква да доведе до затруднение при решаването ѝ.



Фигура 1. Пример за рекурентни редици в първи клас от (Garcheva & Manova 2016)

Характерно за задачите при най-малките ученици е, че зависимостите се представят изцяло нагледно, а от децата се изисква единствено попълване на краен брой стойности, без подробно обяснение. За целта има точно определени квадратчета, в които се попълват търсените числа. При втората редица има потенциален проблем, че ако се продължи с търсене на по-нататъшни членове, би се достигнало до отрицателни числа, а такива не се изучават в първи клас. На тази възраст е изключителна рядкост наличието на любопитство за това какво се случва извън поставените ограничителни рамки в задачите. Поради тази причина въвеждането на подобни условия е допустимо и се приема като далечна пропедевтика – цели се само и единствено да се добие усет за зависимостите, а редиците винаги са крайни (спира се до попълване на последното квадратче).

Подобни задачи се появяват сравнително рядко, но все пак регулярно през всички класове от начален и прогимназиален етап. Например на фиг. 2 е показана задача от учебник за V клас (Vitanov et al. 2016), в която е зададена редицата $a_n = a_{n-1} + 2{,}4$ с първи член $a_0 = 1{,}6$.

11. Всяко следващо число в редицата 1,6; 4; 6,4; 8,8... се получава, като прибавим 2,4 към предходното. Намерете следващите три числа в редицата.

Фигура 2. Пример за рекурентна редица в пети клас от (Vitanov et al. 2016)

Подходът, при който зависимостите се показват първоначално нагледно, а след това се зададени явно чрез подробен текст, се счита за нормален с оглед на това, че учениците все още нямат изградени навици да решават подобни задачи. Следващият етап е зависимостите да се зададат неявно, а учениците да ги преоткрият. Пример за такава задача в VI клас са редиците от (Argirova et al. 2020), показани на фиг. 3. В случая а) учениците трябва сами да намерят зависимостта $a_n = \frac{a_{n-1}}{-2}$, а в случая б) – зависимостта $a_n = \frac{a_{n-1}}{-3}$.

4 Продължете редицата, като напишете още три числа. Разделете първото число на последното.

а) – 6; 3; $-\frac{3}{2}$; ... ; б) 24,3; – 8,1; 2,7; ...

Фигура 3. Пример за рекурентна редица в шести клас от (Argirova et al. 2020)

Този пример следва да бъде разгледан и от гледната точка на неговата коректност. Трябва да се отчете, че условието на задачата от фиг. 3 е зададено неправилно от методическа гледна точка. Зададени са две задачи, които трябва да се приложат поотделно към всяка от двете подзадачи. Втората част от условието (*разделете първото число на последното*) би следвало да се оформи в отделна подзадача или да се премахне напълно. В противен случай някои ученици могат да се заблудят, че този текст има отношение към намирането на рекурентната зависимост.

Може да твърдим, че такива задачи са зададени некоректно от математическа гледна точка. Наистина всяка редица от няколко числа може да се продължи по различни начини, когато не е казано какво е правилото за продължаването ѝ. Например професионален математик може да предпочете решение с интерполационен полином. Такъв за редицата от а) е $f(x) = -\frac{27}{4}x^2 + \frac{117}{4}x - \frac{57}{2}$, при който $f(1) = -6$, $f(2) = 3$ и $f(3) = -\frac{3}{2}$. Тогава търсеното четвърто

число ще е $f(4) = -\frac{38}{2}$, което е различно от очакваното от авторите на учебника $\frac{33}{44}$. Все пак на този етап от обучението учениците не са изучавали интерполационни методи, т.е. от тях се очаква само и единствено интуитивното (може да се нарече и естествено) решение на задачата, поради което подобни условия на задачи могат да се приемат за допустими.

Въпросът с ранното въвеждане на числови редици в прогимназиален етап е проучван подробно от Тони Чехларова. В (Chehlarova 2001a) е показана дидактическа система от задачи с пропедевтика за числови редици, като на базата на нея се прави заключение, че „…*включването на задачи с числови редици в учебното съдържание по математика в VI клас ще допринесе за по-доброто усвояване на знанията и усъвършенстване на уменията на учениците по темата „Рационални числа“, за развиване на мисленето и въображението им, за изграждане на любов към математиката“*. Подобна система е разработена за V клас в (Chehlarova 2001b). В (Chehlarova 2002a) се казва, че „… *разкриването на закономерности съдейства значително за развитието на интелекта на учениците – формират се знания и умения за извършване на наблюдение, сравнение, анализ, синтез, обобщение, аналогия*“. В (Chehlarova 2002b) се изтъква, че подобни задачи са с изследователски характер и по този начин се стимулира творческото мислене на учениците. Въпреки тези усилия към днешен ден споменатата пропедевтика не е силно застъпена в учебните програми. Най-вероятната причина е липсата на достатъчен хорариум. За сметка на това такива задачи се срещат често по време на подготовката на ученици за състезания и олимпиади.

3. Изучаването на числови редици в X клас

Систематичното изучаване на числови редици се извършва в темата *Прогресии* в X клас. Авторите на учебници традиционно подхождат по следния начин:

– въвежда се формално понятието *числова редица* и се показват описателни примери, на базата на които се дефинира рекурентно задаване на общ член и задаване чрез функция, представена с формула в затворен вид;

– въвежда се понятието *монотонна редица*, произхождащо съответно от понятията *растяща* и *намаляваща редица*;

– въвеждат се понятията *крайна редица* и *безкрайна редица*;

– въвежда се понятието *аритметична прогресия* с общ член $a_n = a_{n-1} + d$ с даден първи член a_1 за $n > 1$, която е с рекурентно задаване;

– чрез сумиране на уравненията за първите n члена на аритметична прогресия се извежда представяне от вида $a_n = a_1 + (n-1)d$, което авторите наричат *формула за общ член*;
– извеждат се поредица от свойства на аритметичната прогресия и формулата за сбор на първите n члена;
– преминава се към изучаването на геометрична прогресия, в която се следва същият логически ход както при аритметичната – първо се дават описателни примери, след това се извежда рекурентна формула и формула за общия член, извеждат се свойства и формула за сбор на първите n члена;
– показват се примери за комбинирани задачи, в които се използват едновременно аритметична и геометрична прогресия;
– показват се приложения на аритметична и геометрична прогресия съответно за изчисляване на *проста лихва* и *сложна лихва*.

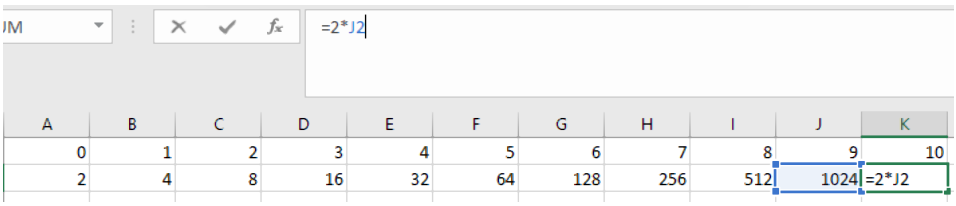
Пример за съвременен учебник, който следва така зададената последователност, е (Bankov et al. 2019). Учебното съдържание практически е строго фиксирано от учебната програма, поради което учебниците на различните издателства нямат съществени разлики в подредбата на учебния материал и в самото съдържание на въпросната тема. Прави впечатление, че учебниците от последните години включват малко задачи за математическо моделиране. В по-ранни учебници, като например (Lozanov et al. 2001) и (Paskalev & Paskaleva 2001), могат да се забележат значително повече практически примери за приложения на рекурентни редици при преподаването на биология, физика, химия и др. В програмата на Международния бакалореат (IBO) математическото моделиране също е силно застъпено, поради което в одобрените учебници като (Haese et al. 2014) се наблюдават множество примери за практическо приложение на рекурентни редици.

4. Изучаването на числови редици по информационни технологии

Двупосочни междупредметни връзки между учебните предмети *математика* и *информационни технологии* биха могли да бъдат изградени лесно по време на изучаването на електронни таблици. Учителите по математика могат да използват средите Microsoft Excel, OpenOffice Calc или LibreOffice Calc, за да спестят голямо количество аритметични пресмятания. Например може да се започне от следната задача:

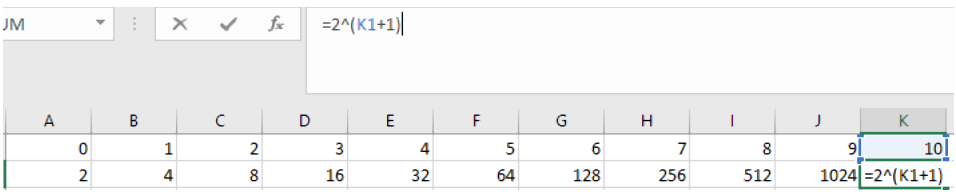
Задача 1. *Намерете* a_{10} *за редицата* $a_n = 2a_{n-1}$, *при* $a_0 = 2$.

Преди да се пристъпи към математическото решение за намиране на общия член, може да се намери решение чрез рекурсивна формула (с относително адресиране на клетките) в приложение за работа с електронни таблици, както е показано на фиг. 4.



Фигура 4. Решение на задача 1 чрез рекурсия в електронна таблица

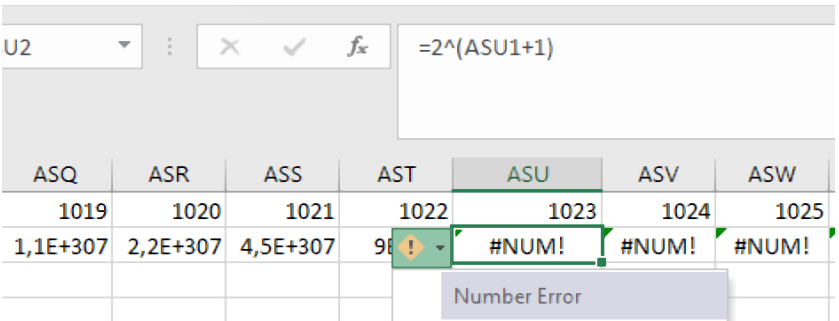
След представяне на общия член като $a_n = 2^{n+1}$ решението може да се визуализира отново чрез формула с относително адресиране, но този път зависещо не от предходния член, а от индекса на елемента. Това решение е показано на фиг. 5.



Фигура 5. Решение на задача 1 чрез формула за общия член

Трябва да се има предвид, че използването на електронни таблици за преподаване на рекурентни редици има основна цел евентуално да се спестят пресмятания и да се използват за проверка на верността на решения, но в никакъв случай не трябва да е заместител на чисто математическото решение на задачата.

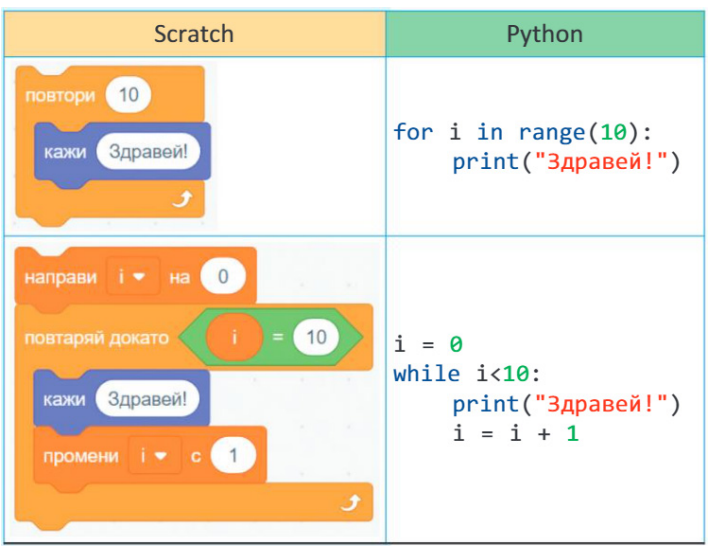
Задачи с рекурентни редици биха могли да бъдат използвани и от учителите по информационни технологии, за да бъдат демонстрирани някои ограничения на програмите за разработка на електронни таблици. Например при решаване на същата задача, лесно се забелязва, че Microsoft Excel ще се окаже в невъзможност да изчисли 2^{1024} (фиг. 6). Грешката се дължи на факта, че $2^{1024} > 9.99999999999999E + 307$, което е най-голямото положително число, поддържано от програмата.



Фигура 6. Невъзможно изчисление на 1023-тия член на редицата от задача 1

5. Рекурентни редици в програмирането

Рекурентните редици се използват в обучението по информатика от много ранен етап. Това се случва още когато учениците започват да се занимават с цикли в часовете по компютърно моделиране в прогимназиалните класове. Там, разбира се, задачите са изключително опростени и редиците може да се приемат като елементарни. Всъщност почти винаги се използва краен вариант (с фиксирана максимална стойност на n) на редицата $a_n = a_{n-1} + 1$, при $a_0 = 0$ (познат като *итератор*). Пример от (Parvanov & Bonev 2022) е показан на фиг. 7.



Фигура 7. Цикли от компютърно моделиране в VI клас

В часовете по информатика в VIII клас основно се използват отново прости итератори, но епизодично е възможно да се появяват и по-сложни задачи с цикли, при които стъпката на цикъла и условието за прекратяването му са по-сложни. Например един от най-сложните алгоритми от (Manev et al. 2017) е показан на фиг. 8.

```
1) нека eps = 0.001
2) въведи a
3) x = a/2.0
4) повтаряй
5) xs = x
6) x = (x + a/x)/2
7) докато (xs - x) >= eps
8) изведи x
```

Фигура 8. Алгоритъм за намиране на квадратен корен

В учебника е посветена цяла страница словесно описание на решението на задачата от фиг. 8. Очевидно е, че авторите целят да се извърши пропедевтика за изучаване на числени методи. Основен проблем е, че на този етап рекурентни редици не са изучавани подробно в часовете по математика и поради тази причина реално няма как такива знания да се използват чрез междупредметна връзка. Авторите на учебника са се опитали да компенсират това чрез описание на достъпен език на понятията *рекурентна зависимост*, *числен метод* и *точност*, но е пределно ясно, че не може да се очаква с такъв кратък прочит и при липса на упражнения те да бъдат осъзнати в дълбочина. С този пример се акцентира, че употребата на по-нестандартни рекурентни редици в обучението по информатика в VIII клас е силно ограничена поради недостатъчното им изучаване в часовете по математика.

От друга страна, изключително подходящи задачи за въвеждане и използване на рекурентни редици в училищния курс по информатика са тези, в които се използва рекурсия. Това може да се направи предварително в IX клас или паралелно с уроците по математика за рекурентни редици в X клас. Въвеждането на рекурсия на по-ранен етап обикновено се избягва, макар че авторите на настоящата статия са на мнение, че някои по-прости класически алгоритми биха били достъпни дори в VIII клас.

Например може да се разгледа следната задача за намиране на n-ти член на рекурентната редица на Фибоначи $a_n = a_{n-1} + a_{n-2}$, за $a_0 = 0$ и $a_1 = 1$. Примерна реализация в програмния език Java би изглеждала по следния начин:

```
static int fib(int n) {
  if (n <= 1) return n;
  return fib(n - 1) + fib(n - 2);
}
```

Този пример може да бъде използван за демонстриране на множество технологични ограничения и проблеми, с които учениците да бъдат поставени в проблемна ситуация. Например извикването `fib(46)` ще даде верен резултат `1836311903`, но извикването `fib(47)` ще даде в резултат некоректно отрицателно число – `1323752223`. Причината за това е надхвърлянето на максималната положителна стойност за тип данни int, която е `2147483647`. Тази задача може да бъде поднесена като софизъм с предизвикателство учениците сами да проучат въпроса „защо се получава така“. След известно асистиране от страна на учителя може да се намери решение чрез библиотеката java.math. BigInteger по следния начин:

```
static BigInteger fib(int n) {
    if (n <= 1) return BigInteger.ONE;
    return fib(n - 1).add(fib(n - 2));
}
```

Рекурсивните алгоритми имат предимството да се реализират с кратък, ясен и елегантен програмен код, но често са неефективни. Такъв е и показаният пример, който в (Koleva & Bakoev 2015) е цитиран като „… *класически пример за неефективна рекурсия*“. Бързо ще се забележи, че подобно решение работи вярно, но е непрактично, защото неговата сложност е експоненциалната $\Theta(\varphi^n)$. Дори изчислението на първоначално търсеното `fib(47)` ще се окаже тежко за обикновен персонален компютър и ще отнеме няколко секунди на стандартен компютър в училищен кабинет. При по-големи числа такова решение е неизползваемо. Тази задача може да се използва непосредствено, за да се покажат техники за оптимизиране на компютърните програми. Например тук е удачно място за демонстрация на *мемоизацията* (Bakoev 2010). Идеята е всяко вече намерено число на Фибоначи да се съхранява, а не да се преизчислява отново и отново. Това може да се реализира чрез масив за пазене на вече изчислени стойности:

```
static BigInteger[] memoArray;
public static void main(String[] args) {
  int n = 1000; memoArray = new BigInteger[n+1];
  System.out.println(fib(n));
}
static BigInteger fib(int n) {
  if (n <= 1) return BigInteger.ONE;
  if(memoArray[n]!=null)return memoArray[n];
```

```
    BigInteger current = fib(n-1).add(fib(n-2));
    memoArray[n] = current;
    return current;
  }
```

По този начин ще могат да се изчисляват изключително бързо големи числа на Фибоначи, защото сложността на алгоритъма вече е линейната $\Theta(n)$. Все пак, следва да се отбележи, че това решение на задачата трябва да се разглежда само като пропедевтика на алгоритмичната техника динамично програмиране. Не може и не трябва да се цели изчерпателност на такива знания в училищния курс. Затова подобни примери следва да се приемат само като възможност да се възбуди интерес към програмирането у по-изявените ученици, а не като нещо задължително, което трябва да се усвои на всяка цена от всички.

Разбира се, стига учениците да имат достатъчно натрупани знания, могат да се упражнят и по-гъвкави структури от данни, като например динамичен масив или хеш-таблица. Подобна адаптивност в задачите (да бъдат надграждани с допълнителни предизвикателства) е изключително ценна в случаите, когато учителят работи с нехомогенна паралелка, т.е. когато в групата има както изявени ученици, така и други, които изостават. При такива ситуации даването на допълнителни задачи за по-напредналите е удобен инструмент за ангажиране на тяхното внимание и предотвратява неприятната ситуация, в която те бездействат, докато чакат останалите да наваксат темпа им.

6. Мотивиране на търсенето на решения с итеративни алгоритми

Практическото приложение на рекурентни редици в програмирането може и трябва да се използва за онагледяване на ползата от търсене на решения с изразяването на a_n като функция на n от часовете по математика. Удачно е учениците да бъдат въведени в проблемна ситуация, която да онагледи предимството на итеративните алгоритми пред рекурсивните. Например рекурсивното решение на задачата за намиране на конкретна степен на числото 2, би изглеждало по следния начин:

```
static BigInteger twoPow(int n){
   if(n < 1) return BigInteger.ONE;
   return BigInteger.TWO.multiply(twoPow(n-1));
 }
```

Въпреки, че то се изпълнява сравнително бързо и ще се получава почти мигновен резултат, ще се забележи, че при големи стойности на n се препълва стекът и възниква java.lang.StackOverflowError. Проблемът се корени в големия брой рекурсивни извиквания на метода. Итеративното решение за този пример ще покаже защо е по-добре рекурсията да се избягва:

```
static BigInteger twoPow(int n){
   BigInteger two = BigInteger.TWO;
   BigInteger result = BigInteger.TWO;
   for(int i=1; i<n; i++) result = result.multiply(two);
   return result;
 }
```

Това не само показва нагледно, че рекурсията по принцип е неефективна, но и мотивира използването на знанията от часовете по математика, при които се търси формула за общия член на рекурентни редици. Също така се вижда, че информационните технологии невинаги могат чрез „груба сила“ да бъдат заместител математическите решения на задачи. Това е силна мотивация за полезността на математиката и показва явно, че един добър програмист трябва да бъде и добър математик.

Същата задача може да се надгради, като се използва за демонстриране на алгоритъма за бързо степенуване. Знаейки, че $2^0 = 1$, $2^n = 2^{n/2}.2^{n/2}$ при четно n и $2^n = 2^{n/2}.2^{n/2}.2$ при нечетно n броят на извършените изчисления може да се съкрати значително. Рекурсивната реализация ще изглежда по следния начин:

```
static BigInteger twoPowSqAndMul(long n) {
  if (n == 0) return BigInteger.ONE;
  BigInteger half = twoPowSqAndMul(n / 2);
  BigInteger squared = half.multiply(half);
  if ((n % 2) == 0) return squared;
  return squared.multiply(BigInteger.TWO);
}
```

При тази задача би могло да се отиде и още по-далеч, като се загатнат някои основи на информатиката и се демонстрира как изучаваните знания за двоична бройна система могат да бъдат изключително полезни. Следното решение демонстрира това:

```
 static BigInteger twoPow(int n){
   return BigInteger.ONE.shiftLeft(n);
 }
```

То, освен че изглежда елегантно заради краткостта си, минимизира употребата на памет и достига до решение с линейна сложност (поради спецификите при реализацията на класа BigInteger). Ако се работеше с примитивен тип данни, можеше сложността да бъде дори константна. Подобни примери разширяват кръгозора на учениците, като ги запознават в по-голяма дълбочина с основите на информатиката, но от чисто практична гледна точка, а не чрез „суха теория“.

7. Рекурентните редици в профилирана подготовка

При преподаване в профилирана подготовка по математика може да се пристъпи към решаване на по-амбициозни задачи, които да разширят знанията за прилагане на рекурентните редици извън тяхната употреба при аритметични и геометрични прогресии. Подходящи са например комбинаторни задачи, които могат да се моделират чрез линейни хомогенни редици. Следният пример е показан в (Griffiths 2011), (Pishro-Nik 2014) и (Doerr & Levasseur 2021):

Задача 2. *Намерете броя на редиците с дължина n, съставени от резултатите от хвърляне на монета със страни ези (E) и тура (T), такива, че в тях да не присъства два пъти последователно ези.*

Търсенето на решение е целесъобразно да започне по индуктивен път за формиране на хипотеза. Учениците могат да намерят първите няколко члена чрез пълно изчерпване. Нека a_n е общият брой. Тогава:

- $n = 1 \rightarrow E$ или T; → отг. 2;
- $n = 2 \rightarrow ET, TE, TT, \sout{EE}$; → отг. 3;
- $n = 3 \rightarrow ETE, TET, TTT, \sout{EEE}, \sout{EET}, TTE, ETT, \sout{TEE}$; → отг. 5;

От първите три члена се вижда, че ако редицата започва с T, на второ място може стои E или T, т.е. има $(n-1)$ позиции. Пример с ограничен брой хвърляния е разгледан в (Griffits 2011). В случай че на първо място се е паднало E, за да бъде изпълнено условието на задачата, на второ трябва да стои т.е. има заети 2 и свободни $(n-2)$ позиции. Множеството на всички редици с n елемента от разглеждания вид е обединение на множеството на всички редици, започващи с T, и на множеството на всички редици, започващи с ET. Броят на редиците в това множество е сумата от бройките в множеството на всички редици от разглеждания вид от $(n-1)$ елемента плюс бройките в множеството на всички редици от разглеждания вид от $(n-2)$ елемента:

$$T\underbrace{\ldots\ldots\ldots}_{n-1} \text{ или } ET\underbrace{\ldots\ldots\ldots}_{n-2}$$

Рекурентната зависимост за тази рекурентна редица е $a_n = a_{n-1} + a_{n-2}$, $n \geq 2$ с начални условия $a_1 = 2$ и $a_2 = 3$. Това очевидно е редицата на Фибоначи (f_n), изместена с един член наляво, т.е. $a_n = f_{n+1}$. Намирането на стойността на a_n за конкретна стойност на n може да се направи с помощта на информационните технологии или с рекурсивна програма, както беше описано в т. 5. Предизвикателството за часовете по математика в профилирана подготовка ще бъде намирането на формулата за общия член, която да доведе до реализирането на итеративен алгоритъм за часа по информатика.

Задачата за намиране на общия член на редицата на Фибоначи може да се приеме като класическа при изучаването на рекурентни редици. Може да се

приложи методът на заместването (Bakoev 2014, p. 175) и да се направи предположение как би изглеждала формулата за a_n като функция на n. Това не е подход, който би могъл да се приложи в общия случай, защото в конкретната задача генераторите на всички редици, удовлетворяващи даденото рекурентно отношение, са добре известни и остава да се намерят числовите коефициенти, отговарящи на началните условия. Решението ѝ е чрез линейно хомогенно рекурентно уравнение, от което се извежда формулата, открита от Абрахам дьо Моавър през 1730 г. (Martinenko 2010), но по-позната днес като *формула на Жак Бине* или *формула на Габриел Ламе*. Двамата математици са я преоткрили независимо един от друг през 1843 г. и 1844 г. (Benjamin et al. 2000).

Преди да се пристъпи към решаването на задача 2, е нужно да се въведат няколко нови понятия и се докажат съответни твърдения.

Дефиниция 2. *Линейно хомогенно рекурентно отношение от ред* k се нарича следното рекурентно отношение $a_n = f(a_{n-1}, a_{n-2}, \dots, a_{n-k}) = c_1.a_{n-1} + c_2.a_{n-2} + \cdots + c_k.a_{n-k}$, където $\forall k \in \mathbb{N}: c_k \neq 0$.

Нека например е зададено следното хомогенно рекурентно отношение от първи ред: $a_n = c.a_{n-1}$. Общият член на съответната рекурентна редица може да бъде намерен по следния начин:

$$a_1 = c.a_0; a_2 = c.a_1 = c^2.a_0; a_3 = c.a_2 = c.c^2.a_0 = c^3.a_0; \dots;$$
$$a_k = c^k.a_0, \dots$$

Общият член на редицата може да се представи във вида $a_n = c^n.a_0$, т.е. $a_n = c^n$ е решение на $a_n = c.a_{n-1}$ при $a_0 = 1$.

За общия случай може да предположим, че за някое *x* решението на хомогенното линейно рекурентно отношение:

$$a_n = c_1a_{n-1} + c_2a_{n-2} + \cdots + c_ka_{n-k},$$

е $a_n = x^n$. Т.е. (след заместването) търси се такова *x*, че:

$$x^n = c_1x^{n-1} + c_2x^{n-2} + \cdots + c_kx^{n-k} \qquad (1)$$

Преобразуваме (1), като прехвърляме всички членове отляво и изваждаме пред скоби общия множител x^{n-k}:

$$x^{n-k}(x^k - c_1.x^{k-1} - c_2.x^{k-2} - \cdots - c_k) = 0 \qquad (2)$$

И така, решения на (2) са тривиалните решения на $x^{n-k} = 0$ и решенията на

$$x^k - c_1.x^{k-1} - c_2.x^{k-2} - \cdots - c_k = 0.$$

Дефиниция 3. *Уравнението* $x^k - c_1.x^{k-1} - c_2.x^{k-2} - \cdots - c_k = 0$ *се нарича характеристично уравнение на линейното хомогенно рекурентно отношение от k-ти ред, а кореният му – характеристични корени на отношението*.

За да бъде намерен общият член на рекурентната редица от задача 2, е достатъчно да бъде разгледан само частният случай, при който $k = 2$ и $D > 0$, където D е дискриминантата на характеристичното уравнение (т.е. уравнението има два различни реални корена).

Лема 1: *Нека A и B са реални числа. Рекурентното отношение*

$a_k = A.a_{k-1} + B.a_{k-2}$, $k \geq 2$, $k \in \mathbb{N}$

се удовлетворява от числовата редица $1, t, t^2, \ldots, t^n, \ldots$, *където* $t \neq 0 \Leftrightarrow t$ *е решение на уравнението* $t^2 - A.t - B = 0$.

***Доказателство*:**

$\Rightarrow$ Допускаме, че за някакво число $t \neq 0$, редицата $\{1, t, t^2, \ldots.\}$ удовлетворява рекурентното отношение $a_k = A.a_{k-1} + B.a_{k-2}$, $k \geq 2$. Това означава, че всеки член на редицата може да се представи като $t^k = A.t^{k-1} + B.t^{k-2}$ и съответно при $k = 2$ се получава

$$t^2 - A.t - B = 0 \qquad (3)$$

$\Leftarrow$ Допускаме, че t е корен на (3). Трябва да отговорим на въпроса дали редицата $\{1, t, t^2, \ldots\}$ удовлетворява $a_k = A.a_{k-1} + B.a_{k-2}$. Ако умножим двете страни на (3) с t^{k-2}, получаваме $t^k = A.t^{k-1} + B.t^{k-2}$. Отговорът е положителен и лемата е доказана.

Оттук може да се докаже и следната теорема:

Теорема 1. *Нека редицата* $\{a_0, a_1, a_2, \ldots, a_n, \ldots\}$ *удовлетворява рекурентното отношение* $a_k = A.a_{k-1} + B.a_{k-2}$, $k \geq 2$. *Ако характеристичното уравнение* $t^2 - A.t - B = 0$ *има два различни реални корена* p *и* q, *общият член на редицата е* $a_n = C.p^n + D.q^n$, *където стойностите на* $C \in \mathbb{R}$ *и* $D \in \mathbb{R}$ *се определят еднозначно от стойностите на* a_0 *и* a_1 *чрез следните уравнения:* $a_0 = C + D$ *и* $a_1 = C.p + D.q$.

Доказателство: използваме метода на математическата индукция.

1. *Проверка за базата*. По условие C и D са дефинирани чрез равенствата $a_0 = C.p^0 + D.q^0 = C + D$ и $a_1 = C.p^1 + D.q^1$, които са изпълнени, откъдето твърдението е в сила за базата.

2. *Индукционно предположение*. Допускаме, че за всяко $k \geq 1$ и за всяко $i \in [0; k]$ е вярно, че $a_i = C.p^i + D.q^i$.

3. *Индукционна стъпка*. Трябва да се докаже, че твърдението е вярно и за $i = k + 1$, или че $a_{k+1} = C.p^{k+1} + D.q^{k+1}$. Това се извършва по следния начин.

Ако $a_k = A.a_{k-1} + B.\text{p}a_{k-2}$, то също така е вярно и $a_{k+1} = A.\ a_k + B.a_{k-1}$ (3).

От индукционното предположение $\Rightarrow$ $a_k = C.p^k + D.q^k$ и $a_{k-1} = C.p^{k-1} + D.q^{k-1}$

Замества се в (3) $\Rightarrow a_{k+1} = A.(C.p^k + D.q^k) + B.(C.p^{k-1} + D.q^{k-1}) \Leftrightarrow$

$$a_{k+1} = A.C.p^k + A.D.q^k + B.C.p^{k-1} + B.D.q^{k-1} \Leftrightarrow$$

$$a_{k+1} = C(A.p^k + B.p^{k-1}) + D(A.q^k + +Bq^{k-1}).$$

От лема 1 $\Rightarrow a_{k+1} = C.p^{k+1} + D.q^{k+1}$. С това теоремата е доказана.

„Доказателствата чрез метода на математическата индукция са едни от най-трудните не само за учениците, но и за студентите от началните курсове.“ (Ganchev et al. 2002, p. 71). Теорема 1 се среща в (Prodanov et al. 1976) и в много други учебници като обикновена задача в контекста на обучението на студенти по математически анализ, но по-скоро ще бъде трудна за ученици. Затова често в училище доказателството на подобни теореми се пропуска. Авторите на статията все пак препоръчват на учителите да експериментират с по-талантливите си ученици. По този повод в рускоезичната литература често се цитират следните думи на акад. Андрей Колмогоров: „*Разбирането на принципа на математическата индукция и умението за правилното му прилагане е добър критерий за логическа зрялост, която е абсолютно необходима за един математик*“.

Изясняването на въпроса какво се случва в този тип задачи, ако корените на характеристичното уравнение са комплексни числа, не следва да се разглежда в училищния курс, защото темата за комплексни числа не е предвидена в учебните програми. Не е невъзможно обаче да възникне въпрос за случая, когато уравнението има двоен корен. Ако характеристичното уравнение има 2-кратен корен, формулата за общия член ще изглежда по следния начин:

$$a_n = C.p^n + D.n.p^n, n \geq 0, n \in \mathbb{N};\ C, D \in \mathbb{R}.$$

Това може да се докаже лесно на база Теорема 1 за $k = 2$, като се вземе предвид, че дискриминантата на характеристичното уравнение е равна на нула и после се направи проверка, че $a_n = n.p^n$ е решение.

Възможен е и компромисен вариант, в който не се извършва изчерпателно доказателство, а се дават само общи евристични насоки. В (Chen 2021) е предложен подобен подход, с който без строго доказателство да се добие усет, че $a_n = n.p^n$ е решение. Разсъждава се в следната посока: ако p_1 и p_2 са два различни, но много близки по стойност корена, може да се покаже, че съществуват такива $C, D \in \mathbb{R}$, за които $a_n = p_2^n - p_1^n$ ще е решение. Трябва да се намери $P \in \mathbb{R}$, такова, че $a_n = P(p_2^n - p_1^n)$ също да е решение. Нека $P = \frac{1}{p_2 - p_1}$, тогава $a_n = \frac{1}{p_2 - p_1}.(p_2^n - p_1^n)$ е решение. Оттук се разглежда границата:

$$\lim_{p_2 \to p_1} \frac{p_2^n - p_1^n}{p_2 - p_1} = \lim_{p_2 \to p_1} \frac{(p_2 - p_1).(p_2^{n-1} + p_2^{n-1}p_1 + \cdots + p_1^{n-1})}{(p_2 - p_1)} = n.p_1^{n-1}$$

и се прави заключение, че $a_n = p_1(n.p_1^{n-1}) = n.p_1^n$ ще е решение. Важно е все пак да се отбележи, че това не е доказателство, а само евристична идея.

Вече може да се пристъпи към по-подробното:

Решение на задача 2. Трябва да се намери формула за общия член на рекурентно зададената редица. По този начин ще се получи общ алгоритъм за намиране броя на редиците, зависещ само от $n \in \mathbb{N}$. За разглежданата рекурентна зависимост полагаме $a_n = x^n$ и заместваме в $a_n = a_{n-1} + a_{n-2}$, $n \geq 2$ **и** $n \in \mathbb{N}$. С деление на $x^{n-2} \neq 0$ получаваме характеристичното уравнение от втора степен с дискриминанта $D > 0$:

$$x^2 - x - 1 = 0 \Longrightarrow x_{1,2} = \frac{1 \pm \sqrt{5}}{2}$$

От Теорема 1 $\Rightarrow a_n = C.x_1^n + D.x_2^n$, където x_1^n **и** x_2^n са генератори на всички редици, удовлетворяващи дадената рекурентна зависимост. За да се реши напълно задачата, е необходимо да се намерят такива $C, D \in R$, че $a_0 = 1$ **и** $a_1 = 2$. Заместваме в общия вид корените на характеристичното уравнение и получаваме:

$$a_n = C.\left(\frac{1+\sqrt{5}}{2}\right)^n + D.\left(\frac{1-\sqrt{5}}{2}\right)^n, n \epsilon \mathbb{N}.$$

Съставяме система от две уравнения с две неизвестни:

$$\begin{vmatrix} C + D = 1 = a_0 \\ C.\left(\frac{1+\sqrt{5}}{2}\right)^1 + D.\left(\frac{1-\sqrt{5}}{2}\right)^1 = 2 = a_1 \end{vmatrix}, \text{ откъдето } \begin{vmatrix} C = \frac{5+3.\sqrt{5}}{10} \\ D = \frac{5-3.\sqrt{5}}{10} \end{vmatrix}$$

Заместваме получените стойности на C **и** D и получаваме формула за общия член на рекурентната редица:

$$a_n = \left(\frac{5+3.\sqrt{5}}{10}\right).\left(\frac{1+\sqrt{5}}{2}\right)^n + \left(\frac{5-3.\sqrt{5}}{10}\right).\left(\frac{1-\sqrt{5}}{2}\right)^n$$

Показаният метод е универсален, но невинаги е достатъчно разбираем и достъпен за учениците. Затова поне в първата задача е добре да се покаже в обратна посока зависимостта между характеристичното уравнение $t^2 - At - B = 0$ и съответната му рекурентна редица и общия ѝ член. Например изхождайки от характеристичното уравнение $x^2 - x - 1 = 0$, може да се изведе следната последователност:

$$x = 1.x + 0; x^2 = 1.x + 1$$
$$x^3 = x^2 \cdot x = x \cdot (x + 1) = x^2 + x = (x + 1) + x = 2x + 1;$$
$$x^4 = x^3 \cdot x = \ldots = 3x + 2; x^5 = x^4 \cdot x = \ldots = 5x + 3$$
. . .

Наблюдава се ясно, че от дясната страна на равенствата коефициентите пред x и свободните членове са последователни елементи на редицата на Фибоначи. Пресмятайки n-тото равенство:

$$x^n = x^{n-1} \cdot x = \ldots = a_n x + a_{n-1},$$

се показва в явен вид връзката между характеристичното уравнение и съответната му рекурентна редица.

На същата основа може да се формулират и друг вид задачи. Нека например $t_1 = 1$ и $t_2 = 2$ са корени на характеристичното уравнение $t^2 = 3t - 2$, което съответства на рекурентната зависимост $a_n = 3a_{n-1} - 2a_{n-2}$. Нека $C = 2$, $D = 1$. Тогава получаваме:

$$a_n = 2.1^n + 2^n = 2 + 2^n,\ a_0 = 3,\ a_1 = 4.$$

По този начин може да се формулира следната задача, подходяща за часовете по профилирана подготовка по математика в XI клас:

Задача 3. *Докажете, че рекурентната редица с начални условия* $a_0 = 3$ *и* $a_1 = 4$ *има общ член* $a_n = 2 + 2^n$.

Подобни задачи са част от текущо предвидената програма за профилирана подготовка в XI клас и най-често се решават с метода на математическата индукция.

Задача 2, от своя страна, би могла да се използва и в часове по информатика, където учениците да се упражнят в моделиране и реализиране на евристичен алгоритъм. Може да се реши и следната:

Задача 4. *Напишете метод, с който се генерират times на брой поредни редици от len на брой хвърляния на монета със страниези (E) и тура (T), който връща като резултат закръглен до цяло число средния брой на редиците, в които няма два пъти последователно ези (EE).*

Решение: Примерно решение на задачата е следното:

```
static long rrET(int len, int times){
  int noEEs = 0; boolean[] array = new boolean[len];
  Random rand = new Random();
  next:
  for(int i=0; i<times; i++){
  // масивът се запълва с произволни E и T
    for(int j=0; j<len; j++) array[j]=rand.nextBoolean();
```

```
        // обхождане за проверка за ЕЕ
          for(int j=0; j<len-1; j++)
            if(array[j]==array[j+1]&&array[j]==true)
              continue next;
        // ако не е намерено ЕЕ, броячът се увеличава
        noEEs++;
      }
      return (long)Math.round(noEEs*(Math.pow(2,len))/times);
    }
```

В него array е булев масив, при който се приема, че false е тура, а true е ези. Основно затруднението се очаква да дойде не от самия алгоритъм за генериране на редици и съответно броене на тези без два пъти ези (практически той е описан словесно в условието на задачата), а в статистическата обработка преди връщането на резултат. Учениците, евентуално подпомогнати от учителя, трябва да намерят, че търсеният резултат е закръгленият брой на редиците без ЕЕ (`noEEs`), умножен по общия брой на всички възможни редици (`Math.pow(2,len)`), разделено на броя направени тестове (`times`). Съществен момент, на който учителят трябва да постави акцент, е закръглянето с метода `Math.round` – без него, при простото превръщане от тип `double` към `int`, ще се закръгля винаги надолу. Удачно е, също така в тази задача учителят да обърне внимание на учениците, че колкото по-голяма е стойността на `times`, толкова по-вероятно е да се получи верен резултат. Това се извършва чрез поредица от изпълнения на алгоритъма с различни стойности на тази променлива.

Дали въпросният алгоритъм би могъл да се реализира от ученици в профилирана подготовка по математика, зависи основно от натрупаните знания и умения по програмиране от предишни години. Възможно е учителят да покаже програмния код наготово и само да го обясни, а програмата да се използва за експериментиране и генериране на хипотези. В профилирана подготовка по информатика не се очаква да е проблем реализацията на предложената програма, но трябва да се отдели известно време за поне общо представяне на математическата ѝ основа. Представеният пример може да се обсъди и от гледната точка на употребата преход към етикет (еквивалента на оператора goto) – защо е добре да бъдат избягвани и кога са допустими.

Относно пропедевтиката на числени методи (като показаната на фиг. 8), на етап в профилирана подготовка такива задачи вече могат да се считат за достъпни не само за демонстрация, но и за реализация от страна на учениците. Редно е да се подберат такива примери, които да се извеждат от самите ученици на базата на знанията им от училищния курс по математика. Пример за това е следната задача:

Задача 5. *От уравненията* $sin(2x) = 2.sin(x).cos(x)$ *и* $sin^2(x) + cos^2(x) = 1$ *може да се изведе формулата* $sin(x) = 2.sin\left(\frac{x}{2}\right)\sqrt{1 - sin^2\left(\frac{x}{2}\right)}$, *която ще е валидна за* $x \in [-\pi, \pi]$

$x \in [-\pi, \pi]$. *Също така се знае, че* $sin(0) = 0$, *и при достатъчно малки стойности на* xx *е вярно, че* $sin(x) \approx x$. *Напишете рекурсивна функция, която при подаден ъгъл* x *и число* n *намира приближение на* $sin(x)$ *след* n *на брой приложения на формулата (на последната стъпка методът да връща* x*).*

***Решение*:** описанието в условието, макар и представено в по-нестандартна форма, е на рекурентната редица $a_n = a_{n-1}\sqrt{1 - \left(\frac{a_{n-1}}{2}\right)^2}$, при $a_0 = x$. Пресмятането ѝ може да се реализира с рекурсивен алгоритъм по следния начин:

```
static double calcSin(double x, int n){
  if(n==0) return x;
  double step = calcSin(x/2, n-1);
  return 2*step*Math.sqrt(1-step*step);
}
```

С това сравнително просто и достъпно решение се извършва пропедевтика на по-задълбоченото изучаване на сходимост на редици и приложение на числени методи. Учениците получават усет как са реализирани някои от библиотечните методи, които те иначе охотно използват наготово. Трябва да се вземе предвид, че в тази задача се използват само и единствено добре изучени тригонометрични формули. По този начин става много по-лесно да се осъзнае същината на алгоритъма. С известно асистиране от страна на учителя учениците биха могли да го изведат и самостоятелно.

Една допълнителна задача за по-напреднали ученици е да се обсъди въпросът какво ще се случи извън интервала $[-\pi, \pi]$. Чрез експериментиране на принципа проба – грешка учениците могат сами да достигнат до извода, че реализираният по-горе метод работи вярно в интервалите $[0 + k\pi, \pi + k\pi]$, когато k е четно, и дава резултати с погрешен знак в същите отворени интервали, когато k е нечетно цяло число. Проблемът се дължи на това, че $\sqrt{1 - sin^2\left(\frac{x}{2}\right)} = \left|\cos\left(\frac{x}{2}\right)\right|$, а в програмния код не е предвидено разкриването на модула с отрицателен знак, когато $\cos\left(\frac{x}{2}\right)$ приема отрицателни стойности.

Едно възможно решение на проблема е да се забрани работата извън правилно работещия основен интервал, с което да се упражни хвърлянето на грешка по време на изпълнение:

```
if(x<-Math.PI || x>Math.PI){
    throw new RuntimeException(„x ∉ [- π, π]“);
}
```

За по-универсално решение учителят може да припомни, че функцията $\sin(x)$ е периодична с период $2k\pi$, $k \in \mathbb{Z}$. Оттук учениците биха могли сами да се досетят как да реализират нормализиране на ъгъла към указания интервал – например като многократно добавят или изваждат 2π, докато попаднат в него:

```
while(x>Math.PI){ x=x-2*Math.PI; }
while(x<-Math.PI){ x=x+2*Math.PI; }
```

Друга посока, в която задачата би могла да бъде разширена, е чрез софизъм покрай следното наблюдение – при подаване на стойност $n > 1075$ методът ще започне да връща 0.0 (а при някои по-специфични ъгли това ще се случва и при малко по-малки стойности – например при $x = 0.08279632679495852$ методът ще започне да връща 0.0 при $n > 1070$). При това, при стойности, близки до тази критична стойност на n, методът ще започне да губи и от точността си. Това се получава заради недостатъчната прецизност на тип данни double, защото той е ограничен до най-малка възможна положителна стойност 2^{-1074}. При достигане до това число операцията деление на 2 ще върне резултат 0.0. Аналогично е положението и при отрицателните ъгли.

Оттук учениците могат сами да изследват при какви стойности на n алгоритъмът започва да дава „достатъчно добър резултат“, като за еталон за сравнение се използва вграденият метод `Math.sin`. Така ще добият чувство за понятието *скорост на сходимост*, което е пропедевтика за теми от висшето образование. Малката разлика в отговорите между реализирания метод и този от вградената библиотека, разбира се, ще се дължи на различните интерполационни алгоритми. За по-любопитните може да се покаже, че Java използва библиотеката `fdlibm`, реализирана на езика за програмиране C, където приближаването се прави чрез интерполационен полином от 13-а степен.

8. Заключение

Целесъобразно е в училищните курсове по информатика и информационни технологии да се отделя повече внимание на междупредметните връзки с училищния курс по математика. В частност, рекурентни редици се използват често в часовете по програмиране и при изучаването на електронни таблици, но сравнително рядко се назовават като такива. Синхронизирането на терминологията между двата учебни предмета и изграждането на явни двупосочни междупредметни връзки биха разширили кръгозора на учениците и биха осмислили с практическо приложение някои математически знания, които обикновено се приемат като прекалено теоретични. В този дух авторите на статията смятат, че отделянето на повече часове за разширяване на знанията за рекурентни редици по време на профилирана подготовка би било от съществена полза за учениците и в двете направления (както математика, така и информатика).

Изследването е подкрепено от проект №80-10-61/25.4.2023 г. (Организационни форми за професионална квалификация на педагогическите специалисти в обучението по математика, информатика и информационни технологии) към Фонд „Научни изследвания“ на Софийския университет „Св. Климент Охридски“, 2023 – 2024 г.

ЛИТЕРАТУРА

АРГИРОВА, Т., КОВАЧЕВА, В., МАРЧЕВА, К., ДИМИТРОВА, С., 2020. *Математика за 6. клас*. „Просвета АзБуки“ ЕООД, София.

BAKOEV, V., 2010. The Recurrence Relations in Teaching Students of Informatics. *Informatics in Education – An International Journal*, vol. 9, no. 2, pp. 159 – 170.

БАКОЕВ, В., 2014. *Дискретна математика: множества, релации, комбинаторика*. КЛМН, София.

БАНКОВ, К., ЦВЕТКОВА, И., ПЕТРОВА, Д., НИКОЛОВА, Г., НАКОВ, С., 2019. *Математика за десети клас*. „Просвета – София“ ЕАД, София.

BENJAMIN, A. T., LEVIN, G. M., MAHLBURG, K., & QUINN, J. J., 2000. Random approaches to Fibonacci identities. *The American Mathematical Monthly*, vol. 107, no. 6, pp. 511 – 516.

ЧЕХЛАРОВА, Т., 2001a. Числови редици за шести клас. *Научни трудове на Пловдивски университет „Паисий Хилендарски“*, vol. 38, no. 2. Методика на обучението.

ЧЕХЛАРОВА, Т., 2001b. Пропедевтика на числовите редици в V клас. В: *Сборник на Юбилейна научна сесия 40 г. секция СМБ, Стара Загора*.

ЧЕХЛАРОВА, Т., 2002a. *Съставяне на задачи с числови редици за 4., 5. и 6. клас*. Научни трудове на СУБ – Пловдив, с. 281 – 286.

ЧЕХЛАРОВА, Т., 2002b. Историята на числовите редици в помощ на обучението. В: *Науката, методиката и училището – конфликтни точки, срещи и разминавания*. Смолян. с. 69 – 72.

CHEN, B., 2021. *Discrete Structures. Lecture Notes*. Math2343, The Hong Kong University of Science and Technology.

DOERR, A., LEVASSEUR, K., 2021. *Applied Discrete Structures*. LibreTexts Project.

ГАНЧЕВ, И., НИНОВА, Ю., НИКОВА, В., 2002. *Методика на обучението по математика (обща част)*. Университетско издателство „Неофит Рилски“, Благоевград.

ГАРЧЕВА, Ю., МАНОВА, А., 2016. *Математика за първи клас*. „Просвета – София“ АД, София.

GRIFFITHS, M., 2011. Fibonacci Expressions Arising from a Coin-Tossing

Scenario Involving Pairs of Consecutive Heads. Fibonacci Quarterly, vol. 49, no. 3, pp. 249 – 254.

HAESE, R., HAESE, S., HAESE, M., MÄENPÄÄ, M., HUMPHRIES, M., 2013. *Mathematics for the International Student: Mathematics SL: Third Edition*. Haese Mathematics, Australia.

КОЛЕВА, К., БАКОЕВ, В., 2015. Златното сечение и числата на Фибоначи – синергетични връзки между математика, информатика и музика. *Математика и информатика*, год. 58, бр. 4, София.

ЛОЗАНОВ, Ч., ВИТАНОВ, Т., НЕДЕВСКИ, П., 2001. *Математика 11. клас задължителна подготовка*. Анубис, София.

МАНЕВ, К., МАНЕВА, Н., ХРИСТОВА, В., 2017. *Информатика 8. клас общообразователна подготовка*. Издателство „Изкуства“, София.

МАРТЫНЕНКО, Г. Я., 2010. Формула Де Муавра-Бине. *Академия тринитаризма*, (Эл. № 77-6567, публ. 16175, 27.11.2010).

ПЪРВАНОВ, И., БОНЕВ, Л., 2022. *Компютърно моделиране и информационни технологии 6. клас*. Домино ЕООД, Стара Загора.

ПАСКАЛЕВ, Г., ПАСКАЛЕВА, З., 2001. *Математика за 11. клас – първо равнище*. Архимед, София.

PISHRO-NIK, H., 2014. *Introduction to Probability, Statistics, and Random Processes*. Kappa Research LLC, www.probabilitycourse.com, Chapter 14.

ПРОДАНОВ, И., ХАДЖИИВАНОВ, Н., ЧОБАНОВ, И., 1976. *Сборник от задачи по диференциално и интегрално смятане. Част 1: Функции на една променлива*. Наука и изкуство, София.

ВИТАНОВ, Т., ЛОЗАНОВ, Ч., ДИЛКИНА, Л., ДЖОНДЖОРОВА, И., ТОДОРОВА, П., КАРАДЖОВА, Р., 2016. *Математика пети клас*. ИК „Анубис“ ООД, София.

REFERENCES

ARGIROVA, T., KOVACHEVA, V., MARCHEVA, K., DIMITROVA, S., 2020. *Mathematics for sixth grade*. Prosveta AzBuki LTD, Sofia. [In Bulgarian]

BAKOEV, V., 2010. The Recurrence Relations in Teaching Students of Informatics. *Informatics in Education – An International Journal*, vol. 9, no. 2, pp. 159 – 170.

BAKOEV, V., 2014. *Discrete Mathematics: Sets, Relations, Combinatorics*. KLMN, Sofia. [In Bulgarian].

BANKOV, K., TSVETKOVA, I., PETROVA, D., NIKOLOVA, G., NAKOV, S., 2019. *Mathematics for tenth grade*. Prosveta – Sofia, Sofia. [In Bulgarian]

BENJAMIN, A. T., LEVIN, G. M., MAHLBURG, K., & QUINN, J.

J., 2000. Random approaches to Fibonacci identities. *The American Mathematical Monthly*, vol. 107, no. 6, pp. 511 – 516.

CHEHLAROVA, T., 2001a. Number Sequences for 6th Grade. *Scientific Works of Plovdiv University "Paisii Hilendarski"*, vol. 38, no. 2. Methods of Education. [In Bulgarian]

CHEHLAROVA, T., 2001b. Propedeutics of Number Squences in 5th Grade. In: *Jubilee Scientific Session 40 years SMB, Section Stara Zagora*.

CHEHLAROVA, T., 2002a. Making of Problems with Number Sequences for 4th, 5th and 6th Grades. *Scientific Works of the Union of Scientists in Bulgaria – Plovdiv*, pp. 281 – 286. [In Bulgarian]

CHEHLAROVA, T., 2002b. The History of Number Sequences in Aid of Teaching. In: *Science, Methodology and the School – Points Of Conflict, Meetings and Divergences*. Smolyan. pp. 69 – 72. [In Bulgarian]

CHEN, B., 2021. *Discrete Structures. Lecture Notes*. Math2343, The Hong Kong University of Science and Technology.

DOERR, A., LEVASSEUR, K., 2021. *Applied Discrete Structures*. LibreTexts Project.

GANTCHEV, I., NINOVA, J., NIKOVA, V. 2002. *Methodics of Education in Mathematics (General Part)*. University Press "Neofit Rilski", Blagoevgrad. [In Bulgarian]

GARCHEVA, YU. , MANOVA, A., 2016. *Mathematics for First Grade*. „Prosveta – Sofia" AD, Sofia. [In Bulgarian]

GRIFFITHS, M., 2011. Fibonacci Expressions Arising from a Coin-Tossing Scenario Involving Pairs of Consecutive Heads. Fibonacci Quarterly, vol. 49, no. 3, pp. 249 – 254.

HAESE, R., HAESE, S., HAESE, M., MÄENPÄÄ, M., HUMPHRIES, M., 2013. *Mathematics for the International Student: Mathematics SL: Third Edition*. Haese Mathematics, Australia.

KOLEVA, K., BAKOEV, V., 2015. The Golden Section and the Fibonacci Numbers - Synergetic Relationship Between Mathematics, Informatics and Music. *Mathematics and Informatics*, vol. 58, no. 4, Sofia. [In Bulgarian]

LOZANOV, CH., VITANOV, T., NEDEVSKI, P., 2001. *Mathematics 11th Grade Regular Preparation*. Anubis, Sofia. [In Bulgarian]

MANEV, K., MANEVA, N., HRISTOVA, V., 2017. *Informatics for 8th Grade*. Izkustva, Sofia. [In Bulgarian]

MARTINENKO, G. Ya., 2010. Formula De Muavra-Bine. *Akademia trinitarizma,* (№ 77-6567, publ. 16175, 27.11.2010). [In Russian]

PARVANOV, I., BONEV, L., 2022. *Computer Modeling and Informational Technologies for 6th grade*. Domino LTD, Stara Zagora. [In Bulgarian]

PASKALEV, G., PASKALEVA, Z., 2001. *Mathematics for 11th Grade – First Level*. Archimed, Sofia. [In Bulgarian]

PISHRO-NIK, H., 2014. *Introduction to Probability, Statistics, and Random Processes*. Kappa Research LLC, www.probabilitycourse.com, Chapter 14.
PRODANOV, I., HADJIIVANOV, N., CHOBANOV, I., 1976. *Collection of Problems in Differential and Integral Calculus. Part 1: Functions of Single Variable*. Nauka i Izkustvo, Sofia. [In Bulgarian]
VITANOV, T., LOZANOV, Ch., DILKINA, L., DZHONDZhOROVA, I., TODOROVA, P., KARADZHOVA, R., 2016. *Mathematics for Fifth Grade*. Anubis LTD, Sofia. [In Bulgarian]

RECURRENT SEQUENCES IN THE SCHOOL COURSE OF MATHEMATICS AND THEIR INTERDISCIPLINARY LINKS WITH COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

Abstract. The article discusses the standard course for learning recurrent sequences in the primary, secondary and high-school mathematics. We propose an extension of the curricula with adding recurrent linear homogeneous sequences in the upper high school course. Additionally, we present related problems in computer science and information technology in order to construct interdisciplinary links, aimed at demonstrating the applied side of recurrent sequences as well as motivating the more gifted students for the learning of some specific computer science basics.

Keywords: recurrent sequences; characteristic equation; mathematical induction; interdisciplinary links; recursive function

✉ **Dr. Borislava Kirilova, Assist. Prof.**
ORCID iD: 0000-0002-4916-2233
Researcher ID (Web of Science): CYL-4772-2022
Sofia University "St. Kliment Ohridski"
Faculty of Mathematics and Informatics
5, James Bourchier Blvd.
Sofia, Bulgaria
E-mail: b.kirilova@fmi.uni-sofia.bg

✉ **Dr. Philip Petrov, Assoc. Prof.**
ORCID iD: 0000-0003-4902-6220
Researcher ID (Web of Science): K-6931-2017
Sofia University "St. Kliment Ohridski"
Faculty of Mathematics and Informatics
5, James Bourchier Blvd.
Sofia, Bulgaria
E-mail: philip@abv.bg