

**ЗА СИСТЕМИТЕ ОТ ОПОРНИ ЗАДАЧИ
ПРИ ПОДГОТОВКАТА ЗА УЧАСТИЕ
В ОЛИМПИАДИ ПО ИНФОРМАТИКА
(КОМБИНАТОРНИ ОБЕКТИ И АЛГОРИТМИ)**

Сава Иванов Гроздев, Коста Андреев Гъров

Настоящата разработка е посветена на подготовката на изявени ученици за участие в олимпиади по информатика. Разглеждат се методически аспекти на системите от опорни задачи. Предлага се примерна подсистема от опорни задачи по темата “Комбинаторни обекти и алгоритми”. Разработката е предназначена за преподаватели в кръжоци и школи в областта на информатиката.

I. Въведение. Едно от методическите направления, в което се работи активно в момента, е уточняване на учебното съдържание на олимпиадите по информатика. На базата на съответното учебно съдържание се правят опити за създаване на системи от “опорни” задачи – такива задачи, които състезателят трябва задължително да реши по време на обучението, за да си осигури необходимия минимум от знания и умения за успешно представяне на олимпиади по информатика. През 1998 г. в [5], К. Гъров и В. Петкова разглеждат проблема за съставяне на система от опорни задачи за подготовка на ученици за участие в олимпиади по информатика във възрастовата група от 5-8 клас. Примери за такива системи и подсистеми са дадени в [6], [7] и [8]. Идеята е да се съставят системи от задачи, подредени по нарастваща трудност. Решавайки поставените опорни задачи, ученикът се изкачва по стълбицата на знанията и уменията, необходими за успешно участие в олимпиади по информатика. Разработените системи и подсистеми от опорни задачи дават възможност на състезателя да избере собствен път в подготовката си. Идеите за системите от тип “стълба” са заимствани от С. Гроздев и П. Кендеров, които в [3] предлагат подобни системи за състезателите по математика.

Съществува разлика между олимпиадните задачи по математика и тези по информатика. Всяка математическа задача е носител на информация и в този смисъл е знание, поради което евентуално усвояеният чрез нея инструмент (знание) е средство за решаване на други задачи. При това може да става дума не само за поставена в началото задача, а и за нейна задача-компонента в контекста на [9, с. 101–105]. В задачите по информатика съществена част е решението, или с други думи, съставянето на алгоритъм и съответна компютърна програма. Задачите-компоненти в математиката, т.е. задачите, чиито решения са части от решението на основната задача, могат да се разглеждат като алгоритмични знания. По-точно те са преходни алгоритмични знания. Дали решенията им трябва да се излагат в основното

решение, зависи от конкретната ситуация. Нивото на подробности се избира от самия ученик в случай, че той познава в детайли всички задачи-компоненти заедно с решенията им. Може обаче да се случи ученикът да няма възможност за избор поради факта, че местонахожденията на задачите-компоненти са различни спрямо вида на “зоните на развитие” в контекста на [9, с. 56–65]. Ще се опитаме да изясним механизма.

Съгласно [9, с. 56], *зоната на актуално развитие* за един ученик се определя от неговите съзрели психически функции в даден момент, т.е. от онези психически процеси в мозъка му, които са предметени в различни знания и тези знания са на такова ниво на осъзнаване, че са се превърнали в инструмент за самостоятелно ползване. При това самостоятелността е съществена. От своя страна, *зоната за близко развитие* съдържа съзряващите психически функции, свързани със знания, които не могат да бъдат използвани без външна намеса. Задачите-компоненти на една задача могат да са разположени в различни зони, включително и извън тях. Ученикът е в състояние да изложи пълни решения на онези, които са в зоната му на актуално развитие. Задачите-компоненти от зоната за близко развитие могат да бъдат използвани само като декларативни знания или да бъдат придружени с непълни решения. Останалите задачи-компоненти извън зоните подлежат на откриване и формулиране в процеса на изработване на решение на основната задача. Справянето с този момент гарантира крайния успех. Аналогично на теоремите, задачите-компоненти са инструменти за формиране на нови знания. Тяхното присъствие в арсенала от знания увеличава възможностите, защото по този начин расте броят на използваните инструменти. Очевидно, колкото по-богат е арсеналът на ученика, толкова по-лесно той се справя с поставените задачи. Разбира се, става дума не за безразборно, а за организирано увеличаване броя на елементите в зоните за активно и близко развитие, което означава осъзнаване до необходимата степен на всеки елемент, а за зоната за активно развитие – осъзнаване до степен на произволност и намиране на подходящото място. Тук важи също принципът на използването на вече готовия инструмент, т.е. инструментът не се изработва (формира) всеки път, когато е необходима употребата му. В обикновените производствени дейности съществува аналогична ситуация и в тази връзка ще припомним следния пример, който принадлежи на проф. Иван Ганчев: всеки път, когато трябва да забиеш пирон, дърводелецът не си прави тесла, а използва вече направената.

II. За системите от опорни задачи при подготовката на ученици за участие в олимпиади по информатика. Разликата между олимпиадните задачи по математика и информатика поставя изискването съответните задачи-компоненти по информатика да бъдат отнесени към подготовката за участие в олимпиади, т.е. те да имат образователен и подготвителен характер, както и да се превърнат в неразделна част от учебния процес. Поради тази причина, вместо задача-компонента използваме термина “опорна учебна задача”. Проблемът за оптимизиране на учебния процес въз основа на методологията на системния подход се разглежда от руския учен В. И. Крупич в [14]. Там той разработва изисквания към системите учебни задачи, които се основават на следните принципи на системния подход:

- принцип на цялостност (единност) – обектът се разглежда като нещо цяло;
- принцип на сложност – изисква се да се отчитат взаимодействията на обекта

със средата и вътрешните фактори;

- принцип на организираност – изисква се да се отчита структурната подреденост на обекта;

- принцип на йерархичност – изисква се да се разглеждат не само връзките между елементите на дадено равнище, но и връзките, които съществуват между различните нива на системата.

На базата на тези принципи могат да се формулират следните общи изисквания към системите от задачи в училищната информатика:

1. Системите от учебни задачи трябва да се състоят от конкретни задачи, насочени към постигане на обобщени цели на учебната дейност.

2. Всяка система от учебни задачи трябва да притежава свойството структурна пълнота. Системите от задачи, построени въз основа на системния принцип на цялостност, повишават ефективността на обучението по фактора време. Основните дейности, с които се извършва икономия на учебното време, необходимо за формиране на определени умения и навици на дадено равнище, са:

- а) намаляване на неоправдано големия брой задачи от системата с минимална сложност;

- б) с повишаване на сложността на задачите да се увеличава и техния брой;

- в) задачите в системата да се аранжират по степен на сложност и ниво на проблемност.

3. Всяка система от учебни задачи трябва да осигурява постепенно увеличаване на сложността, а на всяко ниво на сложност и постепенно нарастване на проблемността.

Известно е, че на олимпиадите по информатика учениците решават задачи, като представят работоспособни компютърни програми, написани на езиците Паскал, С/С++ или Бейсик. В “Регламент за провеждане на олимпиадата по информатика” е записано: “Всяко състезание в рамките на олимпиадата се състои в разработване на компютърни програми, които решават задачи предимно с алгоритмичен характер. Проверката на решението на всяка задача се извършва с комплект тестови данни, които не са известни на състезателя. За всяко вярно изпълнение на програмата върху тестов пример се присъжда определено количество точки. Когато при изпълнението на програма върху тестов пример възникнат грешки или тя не завърши за определеното време, нейното изпълнение се прекратява и не се присъждат точки за съответния тест”. Вижда се, че задачите по информатика съществено се разграничават от задачите по математика (както и по другите учебни дисциплини) – те задължително трябва да се изпълняват на компютър. Ето защо към системите от задачи за подготовка на ученици за участие в олимпиади по информатика се поставя задължителното условие те да съдържат начална подсистема за усвояване на език за програмиране. Създаването на такава подсистема не е елементарна дейност поради факта, че доскоро изучаването на програмиране ставаше в Университетите, а днес най-младите участници в олимпиадите по информатика са в 4 – 5 клас на СОУ. (Интересен е фактът, че на МОИ в Загреб, 13-годишен ученик от Беларус спечели златен медал.)

Важно значение за системите от задачи по информатика е коректното формулиране на условията на задачата. Можем да поставим следните изисквания към формулировката на задачите по информатика:

- задачата трябва да бъде описана като последователност от задания, които ученикът трябва да програмира;
- заданията трябва да са точно формулирани (без двусмислици);
- заданията трябва да са съобразени с времето за тяхното изпълнение, като се вземат под внимание възможностите и възрастта на учениците;
- в задачата трябва да са посочени примерни входни данни и съответните изходни резултати.

Решаването на задача по информатика е следващият нестандартен и труден момент от методическа гледна точка. Освен алгоритмичното решение се изисква практическо изпълнение на компютър, което в много случаи е субективно. Можем да предложим следните осем етапа, през които минава решаването на задачи с компютър:

- постановка на задачата;
- построяване на модел на задачата;
- разработка на алгоритъм за решаване на задачата;
- проверка на верността на алгоритъма;
- анализ на алгоритъма и неговата сложност;
- реализация на алгоритъма с програма на език за програмиране;
- проверка и тестване на програмата;
- съставяне на документация.

Тук ще коментираме само един нестандартен от гледна точка на методиката етап от решаването на задачи на олимпиади по информатика – проверка и тестване на програмата. Както беше отбелязано по-горе, проверката на верността на съставен алгоритъм и съответната му компютърна програма е един от най-трудните етапи при решаване на задачи с компютър. Най-разпространената процедура за доказателство, че една програма е вярна, е подлагането ѝ на различни тестове. Такъв подход е приет и при оценяване на решенията на задачите на ученически и студентски олимпиади и състезания по програмиране. За първи път това става на Втората МОИ, проведена в Минск през 1990 г. Пред състезателите възниква следния основен въпрос: Как да изберем входните данни за тестване? На този въпрос е невъзможно да се даде общ отговор. За всеки алгоритъм отговорът зависи от сложността на алгоритъма, времеви ресурс за тестове по време на състезанието, различните варианти на входни данни и др. Обикновено множеството от всички варианти на входни данни е огромно и пълната проверка е практически невъзможна. Ето защо, трябва да се избере такова подмножество от входове за съответната програма, с които се проверява всеки участък (сегмент) от нея. На проверка подлежат както случаите, които с голяма вероятност се срещат в практиката, така и граничните случаи. Следва да се отбележи, че поне у нас, тестването на дадена сложна програма рядко изключва всички съмнения за съществуването на случаи, в които програмата дава грешни резултати. Разбира се, в горните класове състезателите трябва да бъдат запознати с теоретичните основи на тестването на софтуер. Тези основи са от областта на дискретната математика и учениците трябва да разбират понятия като дискретна функция, релация над декартов квадрат, еквивалентност, граф, таблица на решенията и др. Те трябва да познават и основните правила на изброителната комбинаторика, тъй като създаването на разумни тестови планове е свързано с предварителното оценяване на броя на тестовите случаи. Ето защо, задължителна в

системите от опорни задачи е задачата за съставяне на собствена тестваща програма (т.н. програма – “чекер”).

III. Подсистема от опорни задачи по темата “Комбинаторни обекти и алгоритми”. За пръв път у нас през 1985 г. Асен Рахнев, Коста Гъргов и Огнян Гавраилов в [10] правят опит за определяне на основни теми за подготовка за състезания по програмиране. В по-ново време такъв опит правят Венета Богданова и Галина Момчева в [11]. В тези материали, както и в почти всички ръководства и помагала за извънкласна работа с изявени ученици по информатика, една от основните теми, които се разглеждат, е темата “Комбинаторни обекти и алгоритми”. В настоящата статия е разгледана една примерна подсистема от опорни задачи по тази тема.

Комбинаторните обекти и алгоритми играят изключително важна роля при решаването на редица теоретични и практически проблеми и са в основата на множество алгоритми. В голяма част от задачите, които се решават на олимпиадите по информатика, се изисква намирането на оптимално решение. При решаването на такива задачи често прилаган подход е да се изследват всички възможни допустими решения и от тях да се избере оптимално – едно или няколко. Генерирането (конструирването) на комбинаторни обекти с помощта на комбинаторни алгоритми отговаря точно на този подход. При разглеждане на темата трябва да се имат предвид съответни комбинаторни обекти и алгоритми и да се решат опорни задачи, включващи написване на компютърни програми за:

- генериране, кодиране и декодиране на пермутации на n елемента без и с повторение;
- генериране, кодиране и декодиране на вариации на n елемента от k клас без и с повторение;
- генериране, кодиране и декодиране на комбинации на n елемента от k клас без и с повторение;
- за дадено естествено число n да се намерят всички възможни ненаредени представяния (разбивания) на n като сума от естествени числа (не непременно различни);
- разбиване на множества и представяне на множество като обединение от непразни негов подмножества;
- една редица от положителни цели числа $a_1, a_2, \dots, a_n$ се нарича адитивна верига с дължина n , ако за всяко k , $1 < k \leq n$, съществуват индекси i и j , $1 \leq i \leq j \leq k$ такива, че $a_k = a_i + a_j$; да се състави програма, която за $a_1 = 1$ и въведен последен елемент a_n намира адитивна редица с минимална дължина.

Знанията и уменията, получени чрез горната подсистема, заемат централно място в подготовката на състезателите поради факта, че те се използват активно при решаване на опорни задачи в други подсистеми. Примери за такива подсистеми са: “Абстрактни структури от данни”, “Въведение в теорията на графите”, “Геометрия и програмиране” и др. По тези причини, задачи, използващи комбинаторни обекти и алгоритми, се предлагат за решаване на олимпиади още в прогимназията. Разбира се, някои опорни задачи трябва да се решават няколко пъти в различни възрастови групи. Използваме спираловиден подход в методиката на обучение по информатика и поради факта, че нивото на научното познание в различните възрастови групи е различно.

Ще дадем пример с опорната задача за генериране, кодиране и декодиране на пермутации на n елемента без и с повторение. В прогимназията не е необходимо да се дава строго определение на комбинаторния обект “пермутация”. Достатъчно е с конкретни примери да се изяснят: същността на обекта, различни алгоритми за генериране (образуване) на пермутации, броят им, видовете (без или с повторение). Най-разумно е в тази възрастова група да се програмира рекурсивен алгоритъм за компютърно генериране на пермутациите. Един такъв алгоритъм е следният: поставяме всеки от елементите на първа позиция в пермутацията, след което на останалите $n - 1$ позиции разполагаме последователно всички възможни пермутации на останалите $n - 1$ елемента. Примерен програмен вариант е следната функция *Perm*, написана на езика C++:

```
Void Perm(int k)
{int m;
  if (k>= n) {printPerm (); return; } /* Намерена е пермутация.
                                         Отпечатва се с функцията printPerm*/
  for (m=0; m<n; m++)
    {if (!used [m])) /* ако елементът m не е използван до момента*/
      {used [m]) =1; /* маркираме елемента \textbf{m} като използван*/
        mp [k] = m; /* в масива mp съхраняваме текущото състояние на
                                         пермутацията*/
        perm (k+1); /* рекурсия*/ /* \textbf{if \ldots }*/
        used [m]) = 0; /* размаркираме елемента m като използван*/
      }
    }
}
```

При писането на текста на програмата е разумно да се обърне внимание върху една възможност за приложение на функцията *Perm* при решаване на оптимизационни задачи, които изискват образуване на пермутации. Може да се използва командата *if*, поставена в коментар преди рекурсията. Ако условието за оптималност не може да бъде изпълнено, генерирането на текущата пермутация може да се прекрати и така да се спести компютърно време. От текста на функцията *Perm* се забелязва, че разглеждането на опорните задачи от дадена тема трябва да се **синхронизира** с подсистемата от опорни задачи, посветена на езика за програмиране. От тази подсистема вече трябва да са усвоени знанията и уменията за програмиране на цикли и рекурсивни функции, както и техниките за маркиране и размаркиране на обекти.

В гимназията може да се разгледа един друг важен аспект на задачата за пермутациите. Например, когато се налага на всяка различна пермутация да се съпостави уникално естествено число така, че по-късно да може да бъде възстановена еднозначно от него – кодиране и декодиране на пермутация. Алгоритмите за тези операции са много по-сложни и затова трябва да се изучават в по-късен етап на подготовката. Такива алгоритми са посочени в [12] и [15], където могат да се намерят и съответните програмни кодове.

Решения на някои от опорните задачи от подсистемата “Основни комбинаторни обекти и алгоритми”, задачи могат да се намерят в [12]. Там фигурират и текстове

на задачи, давани на различни олимпиади и състезания, за решаването на които се използват посочените опорни задачи.

IV. Заключение. Ще отбележим, че е направен опит да се определи статистическото разпределение на предлаганите задачи на олимпиади и състезания по информатика в България по теми от учебното съдържание. За целта анализирахме 300 задачи, давани на различни състезания по информатика – Националната олимпиада по информатика; Пролетния турнир по информатика, провеждан в Пловдив и Ямбол; Есенния турнир по информатика, провеждан в Шумен; Зимните състезания по математика и информатика; Коледното състезание по математика и информатика “Димо Малешков” и Великденския турнир по математика и информатика “Румен Грозданов”, провеждани в Пловдив. Оказа се, че решенията на 43 от задачите или 14,33% са **пряко свързани** с опорните задачи от подсистемата “Основни комбинаторни обекти и алгоритми”, което потвърждава нейната значимост при подготовката на изявени ученици по информатика.

ЛИТЕРАТУРА

- [1] С. Гроздев. Олимпиади и синергетика. *Математика и математическо образование*, **32** (2003), 101–116.
- [2] П. КЕНДЕРОВ, С. ГРОЗДЕВ. Европейският модел “МАТНЕУ: Откриване, мотивиране и подкрепа на математическите таланти в европейските училища”. *Математика и математическо образование*, **33**, (2004), 39–49.
- [3] С. ГРОЗДЕВ, П. КЕНДЕРОВ. Инструментариум за откриване и подкрепа на изявени ученици по математика. *Математика и математическо образование*, **34**, (2005), 53–64.
- [4] П. КЕНДЕРОВ, С. ГРОЗДЕВ. Диференцираното обучение по математика в европейските училища. *Математика и математическо образование*, **35**, (2006), 124–134.
- [5] К. ГЪРОВ, В. ПЕТКОВА. Извънкласна работа по информатика в V–VII клас на средното училище. *Математика и математическо образование*, **27**, (1998), 303–312.
- [6] К. ГЪРОВ. Система от опорни задачи при подготовката на таланти ученици за участие в олимпиади и състезания по информатика. *Математика и математическо образование*, **33**, (2004), 316–321.
- [7] К. ГЪРОВ, Е. ГОДОРОВА. Примерна система от опорни задачи по темата “Алгоритми и задачи от теория на числата” за подготовка на таланти ученици по информатика. *Математика и математическо образование*, **35**, (2006), 374–380.
- [8] ГЪРОВ К., В. ПЕТКОВА. Темата “Геометрия и програмиране” в извънкласната работа по информатика. *Математика и математическо образование*, **26**, (1997), 319–325.
- [9] И. ГАНЧЕВ. Основни учебни дейности в урока по математика. ИФ “Модул-96, София, 1999.
- [10] А. РАХНЕВ, К. ГЪРОВ, О. ГАВРАИЛОВ. Ръководство за извънкласна работа по информатика на базата на езика “Бейсик”, изд. на МНП, София, 1985
- [11] В. БОГДАНОВА, Г. МОМЧЕВА. Структуриране на учебното съдържание в извънкласните форми по информатика. *Математика и информатика*, бр.1, 2001.
- [12] П. Наков, П. Добриков. Програмиране = ++Алгоритми, изд. Тор Team Co., София, 2002.
- [13] Р. ТРАШЛИЕВ. Задачата (Психолого-педагогически проблеми). София, 1989.
- [14] В. КРУПИЧ. Теоретические основы обучения решению школьных математических задач. Москва, изд. “Прометей”, 1995.
- [15] Д. ШИШКОВ и колектив. Структури от данни. изд. Интеграл, Добрич, 1995.

Сава Иванов Гроздев
Институт по математика и информатика
ул. Акад. Г. Бончев, бл. 8
1113 София
e-mail: sava.grozdev@gmail.com

Коста Андреев Гъров
бул. Източен 111
4000 Пловдив
e-mail: kosgar@pu.acad.bg

**ON THE SYSTEM OF SUPPORTIVE PROBLEMS
IN THE PREPARATION FOR PARTICIPATION
IN INFORMATICS OLYMPIADS
COMBINATORIAL OBJECTS AND ALGORITHMS**

Sava Ivanov Grozdev, Kosta Andreev Garov

The present paper is devoted to the preparation of outstanding students for participation in Informatics Olympiads. Methodological aspects of systems of supportive problems are considered. An example of a sub-system of supportive problems is proposed, concerning the theme "Combinatorial objects and algorithms". The paper is targeted to teachers of circles and preparation courses in the domain of Informatics.