

УЧНІВСЬКІ ОЛІМПІАДИ З ІНФОРМАТИКИ (СУЧАСНИЙ ЕТАП)

У статті проаналізовано основні задачі проведення олімпіад з інформатики, приділено особливу увагу методиці підбору завдань до олімпіади, особливостям перевірки робіт учасників олімпіади, наведено приклад олімпіадної задачі з розв'язком.

Після здобуття незалежності Україна відкрила нову сторінку у своїй історії. Ставши на шлях якісного оновлення, вона провела ряд реформ, які теж не обминули і освіту. З огляду на потреби економічного та соціального розвитку, гостро постало питання про обдарованість. Тепер це питання набуло державної ваги, в зв'язку з чим і була створена Програма роботи з обдарованою молоддю на 2007-2010 роки. Одним із перших завдань цієї програми є виявлення, розвиток та підтримка обдарованих дітей, учнів та студентів [1]. Виходячи із вищесказаного, проблема навчання обдарованої молоді є досить актуальною та практично значущою на даний час.

Інформаційні технології з кожним роком впевнено впроваджуються в нові сфери діяльності людини. За останні десятиліття комп'ютер із приладу, з яким працювали лише досвідчені програмісти та інженери, перетворився в надійного помічника для багатьох професій. Зацікавленість молоді інформаційними технологіями у третьому тисячолітті різко відрізняється від інтересу до обчислювальної техніки 70-х і навіть 80-х р. минулого сторіччя: крім суто програмістського аспекту поступово з'явилися нові аспекти інформаційних технологій як розв'язування офісних, фінансово-економічних, дизайнерських, інженерно-технічних задач, використання комп'ютерів для обробки звукового та світлового супроводу концертів, створення відео ефектів, розробки 3D-образів у сучасних фільмах. Залежно від природних здібностей кожна особистість вибирає ту з інформаційних технологій, у якій вони можуть найбільш реалізуватися, пробує в цій технології свої сили, приміряє майбутнє професійне життя.

У шкільному курсі інформатики донедавна особливе місце займав курс алгоритмізації та програмування, але за наявності нових програм з інформатики він зайняв позиції розділу, який націлений на розвиток здібностей відповідної категорії учнів – підлітків з розвиненим логічним мисленням, підготовленим математичним апаратом, зацікавлених у формуванні себе як майбутніх програмістів [2].

Олімпіади є одним з об'єктивних показників продуктивності навчальної і позакласної роботи вчителя інформатики. Учителі самі зацікавлені в одержанні таких об'єктивних даних шляхом порівняння рівня знань учнів різних шкіл. Тому вони з інтересом стежать за результатами таких олімпіад.

Традиційно під олімпіадами з інформатики розуміють олімпіади з програмування. Можна вказати на дві основні причини такого вузького розуміння інформатики стосовно олімпіад. Насамперед, у часи "до персональних комп'ютерів" робота на ЕОМ розумілася, насамперед, як програмування. Викладання ж програмування зводилося до вивчення алгоритмів, оскільки алгоритмізація довгий час була найбільш актуальним напрямком програмування. Такий же вузький зміст був закладений у перший шкільний підручник Єршова. Уже перші олімпіади ще в часи безмашинного викладання інформатики були присвячені розв'язанню задач на створення алгоритмів на шкільній алгоритмічній мові. Інерція такого розуміння комп'ютерної справи серед неспеціалістів не переборена і досі, не випадково фахівців з комп'ютерів на підприємствах й сьогодні найчастіше називають програмістами.

Іншою, більш важливою причиною, по якій із усіх розділів інформатики для олімпіад вибирається програмування, є те, що олімпіада по програмуванню універсальна в тому, що відмінності, пов'язані з використанням учасниками різного інструментарію – обчислювальної техніки, мов програмування і середовищ програмування різної потужності – при правильному підборі задач є другорядними в порівнянні з рівнем загального алгоритмічного мислення, досвідом і навичками програміста, знанням можливостей мови програмування, навичками формалізації задач. З цієї точки зору програмування близьке до математики і фізики. Не випадково саме по цих предметах найбільш значні традиції проведення олімпіад на всіх рівнях – починаючи від шкільних і міських (районних), і до міжнародних. Саме універсальність олімпіад з програмування є причиною їх популярності.

Високо оцінюючи значення олімпіад з інформатики, потрібно мати на увазі, що програмування не є основним змістом шкільного курсу інформатики. Тридцять шість годин, відведених для вивчення програмування в шкільних підручниках (програмах) не в змозі навчити школярів ефективного програмуванню. Тому практично школяр не може показати успішні результати на такій олімпіаді, спираючись тільки на знання, отримані на уроках з інформатики. Досвід показує, що справитися з олімпіадною задачею можуть тільки ті школярі, які одержали додаткову підготовку по програмуванню, що значно виходить за рамки стандартної шкільної програми з інформатики.

Учителеві інформатики в будь-якому випадку важливо стежити за підсумками олімпіади, за змістом олімпіадних задач. Від нього не вимагається обов'язково використовувати ці задачі на звичайних уроках зі школярами середніх здібностей. Однак він може зробити деякі загальні висновки і на їхній основі скорегувати зміст курсу інформатики. Наприклад, те, що учасники олімпіади стикаються з труднощами при побудові моделі даних, вказує на необхідність посилення уваги на ці аспекти, причому необов'язково із застосуванням програмування.

Олімпіади взагалі займають особливе місце серед інших організаційних форм навчання. Олімпіади з інформатики мають на меті наступне: пошук обдарованих, розвинутих та освічених школярів; оцінку рівня викладання інформатики у школі в цілому; пред'явлення граничного рівня вимог, орієнтира при вивченні інформатики; реалізацію зворотного зв'язку ланцюга "школа – вуз".

Можна відмітити "занедбаність" деяких учасників олімпіад: їх освіта та розвиток відбуваються стихійно, і навіть деякий матеріал стандартного курсу інформатики їм не знайомий. Ця стихійність виявляється в хитромудрих прийомах при незнанні типових методів розв'язування задач. На простих задачах такі школярі демонструють особливо витончені і "незаконні" витівки, але перед більш складною задачею зупиняється, їх увага направлена не на алгоритмізацію та програмування як особливий вид людського мислення і діяльності і на постановку задач, а на мову програмування. Але відмітимо їх інтуїтивне прагнення до нестандартних методів розв'язування завдань.

Враховуючи основну мету олімпіади – виявлення здібностей школяра, необхідно виконати наступне:

- виявити школярів з розвиненим логіко-алгоритмічним мисленням. Його нерозвиненість може маскуватися використанням готових програмних засобів або бібліотек мови;
- виявити школярів з розвиненим системно-комбінаторним мисленням. Це виявляється в умінні використовувати, причому оригінально та нешаблонно, різноманітні програмні засоби і команди та уникати програмування. Відсутність такої освіченості і типу мислення може маскуватися високим рівнем техніки "голого" програмування.

Олімпіади з інформатики пред'являють свої особливі вимоги до підбору чи розробки задач. Задачі для олімпіади повинні відбиратися з надмірною ретельністю.

Перерахуємо вимоги до олімпіадних задач. По-перше, задача повинна бути невідомою для учасників олімпіади. По цій же причині слід уникати вибору задач з загальнодоступної та популярної літератури з програмування. В ідеалі задачі повинні бути взагалі власної розробки, але для цього потрібна наявність особливих навичок та досвіду в розробників таких задач. Тим паче, розробка власних задач для олімпіади не є непосильною. Річ у тім, що нові задачі можна отримати шляхом переробки відомих задач.

Розв'язання задачі не повинно базуватися тільки на використанні особливих знань. Якщо такі задачі і пропонуються, то тільки серед деяких задач при наявності альтернативних задач з прийнятними для всіх умовами. Олімпіадна задача повинна містити оригінальну ідею, яка вимагає від учасників використання нетрадиційних підходів до її розв'язання. Олімпіадні задачі повинні бути не тільки формально новими по тексту, але й новими для школяра за суттю, за змістовим наповненням, підходом до розв'язування, за методами і засобами, що використовуються. У ході розв'язування такої задачі складністю повинні бути не технічні проблеми відомого методу для цієї задачі, а творчі муки винайдення самого методу.

Якщо ж правильність програм визначається шляхом тестування, від задачі вимагається максимально чітке формулювання, що виключає будь-яку двозначність.

Можна виділити задачі на алгоритм, задачі на структуру даних та задачі на реалізацію на мові програмування. Існує варіант, коли на олімпіаді пропонується не одна, а декілька задач, і кожний учасник має право самостійного вибору. На перший погляд, це дозволяє уникнути провалу для тих учасників, які не знайомі з тематикою і тому не змогли осилити і однієї задачі. Однак у цьому випадку виникають відразу дві проблеми. Перша проблема пов'язана з тим, як оцінювати результати. Практично неможливо передбачити, яка задача виявиться для учасників складнішою або легшою. Тим самим з самого початку ми ставимо учасників у нерівне становище. Крім цього, перед учасниками постає питання вибору задачі. Замість того, щоб зосередитися на розв'язуванні конкретного завдання, вони змушені робити тяжкий вибір, втрачаючи дорогоцінний час. Зменшення числа задач до однієї при одночасному її "розгалуженні" є чіткою стійкою тенденцією.

Бажано підбирати завдання, при розв'язуванні яких програма використовує тільки текстовий режим, оскільки графічні можливості різних типів ЕОМ і систем програмування дуже відрізняються. Задачі повинні бути ще й такими, щоб при розв'язуванні можна було б обійтися засобами, наявними у всіх мовах програмування. В аспекті структур даних до таких універсальних засобів відносяться, крім простих типів даних, масиви та рядкові.

Аналізуючи тематику задач олімпіад різних років, можна відмітити, що відбувається відхід від задач обчислювального характеру. Помітно, що такі задачі в значній мірі потребують наявності математичного типу мислення і тому в деякому розумінні не відповідають олімпіадам з інформатики. З іншого боку, відомо, що на олімпіадах з фізики останніх років почали з'являтися задачі, що потребують використання обчислювальної техніки [3; 4; 5].

Таким чином, виникає природний процес розподілу різноманітних комп'ютерних умінь на ті, які є предметом змагань на олімпіадах з інформатики та на ті, що служать інструментом для інших наук.

В якості спільних тематик олімпіадних задач, де можна підібрати цікаву задачу, можна виділити: перебір на деревах, довга арифметика, шифровка і дешифровка, обробка рядків, синтаксичний аналіз формул і т.д. [6; 7; 8]. В якості приклада пропонуємо задачу, що була використана на обласному етапі шкільної олімпіади з програмування в місті Житомирі.

Від'їжджаючи з домівки, поет залишив коту, прикутого до дуба ланцюгом довжиною L , N рибин. Знаючи координати голови та хвоста кожної з них, поррахуйте, на яку добу у kota виникне почуття голоду, якщо він починає голодувати тоді, коли за добу з'їсть менше, ніж K рибин. Рибину кіт може з'їсти тільки тоді, коли зможе дотягнутися хоча б до однієї її точки. Координати дуба $(0, 0)$. Усі числа цілі, не перевищують за модулем 100. (Автор задачі Присяжнюк А.В.)

Незважаючи на те, що для розв'язку цієї задачі необхідно вміти розв'язувати лише квадратні рівняння, більша частина учасників не спромоглися її розв'язати. Тому наводимо текст розв'язку задачі (програми на мові Pascal) з коментарями.

```
Program cat;
var{опис змінних}
f: text;
l,n,k,x1,y1,x2,y2,i,m:integer;
xa,ya,xb,yb,r:real;
Procedure kv (q,w,e:real); {пошук коренів квадратного рівняння}
var d:real;
begin
  d:=w*w-4*q*e;
  if d>=0 then begin
    if d>0 then d:=sqrt(d);
    r:=2; xa:=(-w+d)/(2*q); xb:=(-w-d)/(2*q);
  end
  else r:=0;
end;
Function ppp:boolean; {}
begin
  ppp:=false;
  if ((x1-xa)*(x2-xa)<0) and ((y1-ya)*(y2-ya)<0) then ppp:=true;
  if ((x1-xb)*(x2-xb)<0) and ((y1-yb)*(y2-yb)<0) then ppp:=true;
end;
Function fish:integer; {перевіряємо, чи можна дотягнутися до риби}
var
  a,b,c,x,y:real;
  flag: boolean;
Begin
  flag:=false;
  if (l!>=x1*x1+y1*y1) or (l!>=x2*x2+y2*y2) then flag:=true
  else begin
    a:=y2-y1; b:=-x2+x1; c:=-b*y1-a*x1;
    if a=0 then begin kv(b*b,0,l!*b*c); ya:=-c/b; yb:=ya; end
    else begin
      kv(b*b+a*a,2*b*c,c-l!*a*a);
      if r=2 then begin ya:=xa; yb:=xb; xa:=(-c-b*ya)/a; xb:=(-c-b*yb)/a; end
    end;
    if r=2 then flag:=ppp;
  end;
  if flag then fish:=1 else fish:=0;
end;
Begin
Assign (f,'a:\cat.dat'); reset(f); readln(f, l, n, k); m:=0;
for i:=1 to n do
  Begin readln(f,x1,y1,x2,y2); m:=m+fish; end;
  m:=(m div k)+1; close(f);
Assign (f,'cat.sol'); rewrite(f); writeln(f,m); close(f);
End.
```

Важливою особливістю олімпіади з інформатики є те, що самі тексти створених програм чи алгоритмів-розв'язань задач, перевіряти набагато важче, ніж розв'язання задач на олімпіадах з математики чи фізики. Досвід показує, що для цього необхідно задіяти більшу кількість перевіряючих-спеціалістів високої кваліфікації з програмування, взаємно контролюючих один одного, та декілька днів роботи.

Виконати перевірку олімпіадних задач шляхом аналізу тексту програми досить важко, навіть якщо вимагати від учасника подання повних коментарів. В умовах нестачі часу учасники квапляться писати програму, а не коментарі, залишаючи їх створення на потім. Часто виходить, що час минув, програма написана, а коментарі не оформлені. Коментарі, достатні з точки зору автора, журі може визнати недостатніми.

1. Іноді перевірка тексту програми взагалі неможлива. Тому зазвичай вихід з цього положення – тестування програм за допомогою набору тестів, без аналізу її змісту. Тестування в таких випадках являється єдиним виходом, хоча слід мати на увазі і негативні сторони такого підходу перевірки правильності розв'язання.

2. Використання тестів не врятує ситуацію повністю, так як, з однієї сторони, важко говорити про їх достатність, а з іншої сторони, виникає запитання: як оцінити програму, якщо тільки частина тестів проходить, або не проходить ні один тест, коли програма містить синтаксичні помилки чи розв'язує тільки частину задачі.

3. Набір тестів повинен охоплювати критичні випадки вхідних даних. У залежності від того, скільки тестів успішно пройшла програма, автор отримує різну кількість балів. У якості додаткових параметрів можуть враховуватись рівень відпрацьованості алгоритму, ефективність розв'язування, врахування граничних та особливих умов, зрозумілість описання програми. Додаткові, заохочуючі, бали можуть бути нараховані за

сервіс, наданий програмою: зручність введення даних, наочність відображення процесу розв'язування та результатів, захист від неправильного введення даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. Програма роботи з обдарованою молоддю на 2007–2010 роки. (08 серпня 2007) [WWW документ]. <http://zakon.rada.gov.ua/cgi-bin/laws/main.cgi?nreg=1016-2007-%EF>
2. http://dl.sumdu.edu.ua/e-pub/ped/_white/ukr/new/1.html
3. Кузічев М.М. I Всеукраїнська олімпіада з інформаційних технологій // Комп'ютер у школі та сім'ї. – 2004. – № 8 – С. 32-34
4. Кузічев М.М. Олімпіада з інформаційних технологій: пошуки, досвід, перспективи // Комп'ютер у школі та сім'ї. – 2004. – № 7 – С. 29-31.
5. Тітова Т.І. Шкільна альтернативна олімпіада з інформатики // Комп'ютер у школі та сім'ї. – 2004. – № 7 – С. 19-22.
6. <http://www.olymp.vinnica.ua/>
7. <http://acm.lviv.ua/>
8. <http://www.e-olimp.com.ua/>

Матеріал надійшов до редакції 25.05.2008р.

Ученические олимпиады по информатике (современный этап)

В статье проанализировано основные задачи проведения олимпиад по информатике, особое внимание уделено методике отбора заданий для олимпиады, особенностям проверки работ участников олимпиады, приведен пример олимпиадной задачи с решением.

Students Olympiad in Informatics (Modern stage)

The article deals with the principle tasks of Olympiads in informatics implementation. Special attention is paid to task choice methodology and peculiarities of participants' works verification. The article suggests an example of an ordinary task with the key.