

Олімпіадні задачі з *інформатики*

Рубан М.М., Устінов С.Є.

Розв'язання задач
II етапу Всеукраїнської олімпіади
з інформатики — 2011 р.

Серія заснована у 2008 році

УДК 37.016:004.021

ББК 22.1

P820

Рецензенти

кандидат фіз.-мат. наук, КОСТИКОВ О.А.,
доцент кафедри прикладної математики,
Донбаської державної машинобудівної академії
кандидат фіз.-мат. наук, КАДУБОВСЬКИЙ О.А.,
Слов'янський державний педагогічний університет,
доцент кафедри ГМВМ

Затверджено

Вченою радою Слов'янського державного
педагогічного університету.
Протокол №08 від 5 квітня 2012 року

Відповідальний за випуск

асистент кафедри алгебри
Рубан М.М.

Рубан М.М., Устінов С.Є.

ОЛІМПІАДНІ ЗАДАЧІ З ІНФОРМАТИКИ: Розв'язання задач II етапу Всеукраїнської олімпіади з інформатики – 2011 р. – Слов'янськ, 2012. – 108 с.: іл. (Випуск 11, Серія: Викладачі СДПУ – учням, студентам, вчителям...)

ISBN 978-966-1554-81-7

Для вчителів, методистів, керівників гуртків з програмування, викладачів та студентів спеціальностей «математика», «інформатика» вищих навчальних закладів.

УДК 37.016:004.021

ББК 22.1

ISBN 978-966-1554-81-7

© Рубан М.М., Устінов С.Є.

Зміст

Вступ	4
Список позначень	6
Класифікація задач	7
Волинська область	9
Дніпропетровська область	15
Донецька область	22
Закарпатська область	38
Запорізька область	43
Івано-Франківська область	57
Київська область	64
Львівська область	71
Миколаївська область	75
Рівненська область	80
Херсонська область	90
Хмельницька область	95
Черкаська область	100
Література	104
Ресурси мережі Інтернет	106

Вступ

*Есть книги, которые надо только отведасть,
есть такие, которые лучше всего проглотить,
и лишь немногие стои́т разжевать и переварить...*
/Фрэнсис Бэкон (Francis Bacon)/

Одним з основних напрямків розвитку змагань з інформатики серед учнів та студентів є поширення різноманітних турнірів зі спортивного програмування. Це і міжнародні турніри, що проходять під егідою міжнародної організації АСМ (*Association for Computing Machinery*), і турніри в рамках держав та їх територіальних одиниць, змагання між студентами університетів, галузевих університетів тощо. Всі ці турніри поєднує єдина форма проведення, та спільна форма організації завдань. Такі турніри вже перешагнули границі очних змагань і все більше існує інтернет-ресурсів, за допомогою яких організуються як індивідуальні змагання так і командні (див. *Ресурси мережі Internet*).

Чудовими матеріалами для підготовки до нових олімпіад є збірники олімпіадних задач, які вже проходили. І хоча з інформатики проведено вже велику кількість олімпіад та турнірів, а з задачами, які там було запропоновано можна ознайомитись на сайтах Інтернету, систематизованих і методично оформлених збірок в навчальних закладах практично не існує.

Даний випуск містить задачі олімпіад II етапу Всеукраїнської олімпіади з інформатики, які проходили в Волинській, Дніпропетровській, Донецькій, Закарпатській, Запорізькій, Івано-Франківській, Київській, Львівській, Миколаївській, Рівненській, Херсонській, Хмельницькій, Черкаській областях в 2011 році. І хоча для 8-9 класів умовою задачі не передбачалось написання комп'ютерної програми, але наведені приклади розв'язання задач будуть дуже корисними.

Тексти задач приведені в тому вигляді, в якому вони пропонувалися на олімпіаді, щоб зберегти повне уявлення про кожну область. До кожної задачі наведено вказівки та програмне рішення.

Основна мета посібника – допомога:

- ✓ вчителю, викладачу, тренеру – системно організувати підготовку учнів та студентів, до олімпіад і турнірів з інформатики;
- ✓ учням, студентам – перевірити рівень своєї підготовки, розв'язуючи завдання із збірника і оцінюючи свої розв'язки;
- ✓ всім учасникам олімпійського руху з інформатики – підвищити свій олімпійський потенціал, ознайомившись із запропонованими розв'язками задач.

Ми використовуємо мову *Pascal* для представлення алгоритмів, які

описуємо, тому, що це одна з найпоширеніших мов програмування. Спочатку алгоритми представляються у вигляді аналізу чи в іншій абстрактній формі. Це зроблено для того, щоб показати весь спектр проблем при розв'язку практичних задач: від проблеми формалізації задачі до проблем, які можуть виникнути під час виконання завершеної програми.

Алгоритми, які ми пропонуємо, можна реалізувати на будь-якій відомій Вам мові програмування.

Зверніть увагу, що всі задачі класифіковані за тематикою, повний список задач розміщено в розділі «Класифікація задач».

Зауваження Після назв деяких задач читач можете зустріти таку позначку: [**EO:X**], яка означає що дану задачу або подібну до неї можна знайти на сайті www.e-olimp.com за номером X (або за посиланням www.e-olimp.com/problems/X).

Бажаємо успіхів!

Список позначень

*Плохая система обозначений может сделать
хорошее изложение плохим, а плохое – еще худшим;
лучшее обозначение – отсутствие обозначений.
/Пол Ричард Халмош (Paul Richard Halmos)/*

<i>Позначення</i>	<i>Опис</i>
$ x $	модуль числа x
$n!$	факторіал числа ($n! = 1 \cdot 2 \cdot \dots \cdot n$, $0! = 1$)
$\sum_{i=1}^n i$	сума чисел від 1 до n
$\sum_{k=1}^n a_k$	сума елементів від a_1 до a_n
$\min(a, b)$	мінімальне число серед чисел a та b
$\max(a, b)$	максимальне число серед чисел a та b
$x = \overline{a, b}$	x пробігає всі цілі значення на відрізку $[a, b]$
$a \div b$	a кратне b
$a \nmid b$	a не кратне b
$\gcd(a, b)$	найбільший спільний дільник (НСД) двох цілих чисел a та b
$\text{lcm}(a, b)$	найменше спільне кратне (НСК) двох цілих чисел a та b
$\lfloor x \rfloor$	найбільше ціле число, яке менше або рівне x : $\max\{n \mid n \leq x\}$
$\lceil x \rceil$	найменше ціле число, яке більше або рівне x : $\min\{n \mid n \geq x\}$
$\text{mod}(a, b)$	остача від ділення: $a - b \lfloor a/b \rfloor$
$\text{div}(a, b)$	цілочисельне ділення: $\lfloor a/b \rfloor$

Класифікація задач

*Есть лишь два вида языков программирования:
те, на которые вечно жалуются, и те,
которые никогда не используются.
/Бьёрн Страуструп (Bjarne Stroustrup)/*

Для зручності в користуванні задачі збірника було розподілено на класи. В дужках записано номер сторінки на якій Ви зможете знайти умову задачі.

Прості задачі

Дві купи (22), Як провести відпустку(15), Пісня (16), Прямокутники (9), Олімпіада (38), Елітне взуття (43), Домашнє завдання (47), Подарунки (57), Подія (57), Євро 2012 (71), Кульки (84), Подорож (91), Сума (95), Переможець (96), Ремонт (100), Послідовність (100), Рибальська сітка (102).

Алгебра і теорія чисел

Рівні відрізки (36), Переміщення дамки (27), Розмін (15), Квадрат (9), Сірники (45), Міська площа (52), Matrix (66), Determ[inant] (68), Бандура (75), Трикутник Василя (76), Улюблені числа автогонщика (77), Проект міста (81), Фактор числа (93), Дільники (95), Точки (99), Дроби (101).

Системи числення

Bin[ary]Fract[ions] (64), Num[bers]Syst[ems] (65), «Щасливий» календар (53).

Геометрія

Ромби (32), Два кола (38), Прямокутник (41), Багатокутник (61), Літаюча камера (73), Спільний відрізок (98).

Перебір варіантів

Рівні відрізки (36), Ромби (32), Обертальні двері (50), Шкаф (80).

Структури даних

Доставка пошти (24), Простий шлях (35).

Сортування та пошук

Третя сума (22), Кращий з гірших vs. гірший з кращих (28), Обробка числа (40), Вболівальниці (71).

Моделювання

Шоу (18), Обробка числа (40), Метод бутерброда (44), Квас (48), Дракон (54), Намисто (58), День народження (84), Вираз (90).

Пошук в глибину (рекурсія)

Білі плями (12), Клумба з квітами (82), Бобри (87).

Динамічне програмування

Зняття табличок (25), Перефарбування паркану (30), Слон (19), Вугілля (86).

Довга арифметика

Слон (19), Ламана (10).

Волинська область

*Хорошо выраженная мысль
звучит умно на всех языках.
/Джон Драйден (John Dryden)/*

Задача 1. Прямокутники. Шоколадну плитку спочатку розламали N разів, потім кожну утворену частину розділили M разів. Визначити загальну кількість утворених прямокутників.

Вхідні дані. Вхідний потік містить єдиний рядок з двох чисел N та M розділених пропуском ($0 \leq N, M \leq 2\,147\,483\,647$).

Вихідні дані. Вихідний потік містить єдине ціле число — кількість прямокутників

Обмеження за часом: 1 сек. на тест.

Приклад вхідних і вихідних даних.

<i>Приклад введення:</i>	<i>Приклад виведення:</i>
1 1	4

Аналіз. Після першого поділу отримаємо $N + 1$ шматок. Далі з кожного шматка вийде $M + 1$ шматок, тобто загальна кількість буде рівна $(N + 1)(M + 1)$.

Програмна реалізація наведеного розв'язку:

```

1 var n,m: Int64;
2 begin
3   Readln(n, m);
4   Writeln((n+1)*(m+1));
5 end.
```

Задача 2. Квадрат. Дано цілі числа N та M , які задають розмір шоколадної плитки. Визначити найменшу кількість частинок квадратної форми на яку можна поділити плитку.

Вхідні дані. Вхідний потік містить єдиний рядок з двох чисел розділених пропуском ($0 < N, M < 2^{63}$).

Вихідні дані. Вихідний потік містить єдине ціле число — кількість квадратів.

Обмеження за часом: 1 сек. на тест.

Приклад вхідних і вихідних даних.

<i>Приклад введення:</i>	<i>Приклад виведення:</i>
2 4	2

Аналіз. Кількість квадратів можна отримати шляхом послідовного відкидання квадратів зі стороною рівною меншій стороні прямокутника. Щоб процес відкидання проходив більш швидко, то рекоменду-

ється замість операції віднімання використовувати операції цілочисельного ділення та остачі від ділення. Даний процес дуже схожий на алгоритм Евкліда (пошук найбільшого спільного дільника двох чисел).

Програмна реалізація наведеного розв'язку:

```

1 var n,m,k: Int64;
2 begin
3   Readln(n, m); k:=0;
4   while (n>0) and (m>0) do
5     if n>m
6       then begin Inc(k, n div m); n:=n mod m; end
7       else begin Inc(k, m div n); m:=m mod n; end;
8   Writeln(k);
9 end.
```

Задача 3. Ламана. Шоколадна плитка являє собою сітку з горизонтальних та вертикальних ліній, точки якої в декартовій системі координат на площині позначено точками з цілими координати. Потрібно поділити шоколадну плитку наступним чином:

- ✓ починати з лівого нижнього кута, який знаходиться в початку координат;
- ✓ можна пересуватися вздовж цих прямих;
- ✓ при проходженні через точку завжди змінювати напрям швидкості на перпендикулярний.

Знайти мінімальну довжину шляху до верхньої правої точки.

Вхідні дані. Дано два числа — розміри шоколадної плитки (цілі числа, не більші 10^{100000}). Числа розділено пропуском.

Вихідні дані. Вивести єдине число — шукану величину.

Обмеження за часом: 1 сек. на тест.

Приклад вхідних і вихідних даних.

Приклад введення:	Приклад виведення:
2 3	5

Аналіз. Відповідь на задачу знаходиться наступним чином: якщо a та b (розміри шоколадної плитки) мають однакову парність, то потрібно знайти $2 \cdot \max(a, b)$ інакше обчислюємо $2 \cdot \max(a, b) - 1$. Але, так як розміри плитки обмежені 10^{100000} , то при написанні програми необхідно використовувати «довгу арифметику».

Програмна реалізація наведеного розв'язку:

```

1 type longByte = array [0..101000] of Integer;
2 var a,b: longByte;
3     z,s: Integer;
4     c: char;
```

```
5
6 procedure longMax(x,y: longByte; var w:longByte);
7 var t:Integer;
8 begin
9   w:=x;
10  if x[0]>y[0] then w:=x
11  else if x[0]<y[0] then w:=y
12      else begin t:=1;
13          while (x[t+z]=y[t+z]) and (t<=y[0]) do inc(t);
14          if x[t+z]>y[t+z] then w:=x
15          else if x[t+z]<y[t+z] then w:=y;
16      end;
17 end;
18
19 procedure longMult(var x: longByte; y:Integer);
20 var t,mem:Integer;
21 begin
22   mem:=0;
23   for t:=z+x[0] downto z+1 do begin
24     mem:=x[t]*y+mem;
25     x[t]:=mem mod 10;
26     mem:=mem div 10;
27   end;
28   while mem>0 do begin
29     Inc(x[0]); x[z]:=mem mod 10; Dec(z);
30     mem:=mem div 10;
31   end;
32 end;
33
34 procedure longMinus1(var x: longByte);
35 var t:Integer;
36 begin
37   if x[x[0]+z]>0 then Dec(x[x[0]+z])
38   else begin
39     t:=x[0]+z;
40     while x[t]=0 do begin
41       x[t]:=9; Dec(t);
42     end;
43     Dec(x[t]);
44     if (x[t]=0) and (t=z+1) then begin Dec(x[0]); Inc(z); end;
45   end;
46 end;
47
48 procedure longPrn(x: longByte);
49 var t:Integer;
50 begin
51   for t:=z+1 to z+x[0] do write(x[t]); Writeln;
52 end;
53
54 begin
55   z:=100;
56   a[0]:=0; b[0]:=0;
```

```

57  read(c);
58  while c in ['0'..'9'] do begin
59      inc(a[0]); a[a[0]+z]:=ord(c)-ord('0');
60      read(c);
61  end;
62  read(c);
63  while c in ['0'..'9'] do begin
64      inc(b[0]); b[b[0]+z]:=ord(c)-ord('0');
65      read(c);
66  end;
67  if a[a[0]+z] mod 2 = b[b[0]+z] mod 2
68  then s:=0 else s:=1;
69  longMax(a,b,a);
70  longMult(a,2);
71  if s=1 then longMinus1(a);
72  longPrn(a);
73  end.

```

Задача 4. Білі плями. Кондитерська фабрика випустила чорно-білий шоколад. Програмістам доручили порахувати вміст білого шоколаду. Співробітники фабрики зберігають інформацію про шоколадну плитку в комп'ютері у вигляді матриці розмірністю $M \times N$. Комірка матриці містить 0, якщо шоколад чорний та 1, якщо білий. У матриці комірки входження білого шоколаду не можуть дотикатися одна до одної ні сторонами, ні кутами. Необхідно написати програму, яка знаходитиме загальну кількість плям білого шоколаду та кількість плям з однаковою площею.

1	0	1	0	0
0	0	1	1	0
1	0	0	0	0
1	0	0	0	1
1	0	1	0	1

Вхідні дані. Вхідний потік містить в першому рядку два числа M та N ($2 \leq M, N \leq 100$), далі слідує M рядків, у кожному по N цілих чисел розділених пропусками — елементи матриці.

Вихідні дані. Вихідний потік містить у першому рядку ціле число k — загальну кількість білих плям, далі у кожному з рядків міститься по два числа, перше — площа плями білого шоколаду, друге — їх кількість. Дані відсортувати за площами, в порядку зростання.

Обмеження за часом: 1 сек. на тест.

Приклад вхідних і вихідних даних.

Приклад введення:	Приклад виведення:
5 5	5
1 0 1 0 0	1 2
0 0 1 1 0	2 1
1 0 0 0 0	3 2
1 0 0 0 1	
1 0 1 0 1	

Аналіз. Це класична задача на використання пошуку в глибину. Зрозуміло, що треба обходити матрицю і якимось чином обчислювати кількість білих плям. Підхід до розв'язку такий: після того, як ми потрапляємо на пляму, треба це зафіксувати, збільшивши змінну-результат на 1. Щоб другий раз не порахувати одну і ту саму пляму, відразу після відвідування необхідно її знищити (стерти), тобто присвоїти всім клітинам плями значення 0.

Для опрацювання плям напишемо процедуру знищення плям, назовемо її *slick* (пляма). Процедура *slick* рекурсивно огляне кожен з суміжних клітин (зверху, знизу, праворуч, ліворуч), та додасть її до плями. Щоб під час виконання процедури не «вискочити» за межі масиву, зробимо його не розміром $M \times N$, а розміром $(M + 2) \times (N + 2)$, це дасть нам можливість оточити вхідний масив розміром $M \times N$ нулями.

Масив $P(10\,000)$ будемо використовувати для збереження кількостей відповідних площ, в P_k -й комірці зберігається кількість плям з площею k . Завдяки такому підходу не потрібно додатково сортувати отримані дані. Чому розмірність масиву 10 000? Тому що M та N зверху обмежені числом 100, отже максимальна можлива площа плями може бути 10 000 (100×100) клітин.

Програмна реалізація наведеного розв'язку:

```

1 var i, j, n, m, t, s: Integer;
2   map: array [0..101, 0..101] of integer;
3   p: array [1..10000] of Integer;
4
5 function slick(i, j: integer): integer;
6 begin
7   if map[i, j] = 0 then island:=0
8   else begin
9     map[i, j] := 0;
10    slick:=1+slick(i+1, j)+slick(i-1, j)+slick(i, j+1)+slick(i, j-1);
11  end;
12 end;
13
14 begin

```

```
15 Readln(m,n); s:=0;
16 FillChar(p, SizeOf(p),0);
17 FillChar(map, SizeOf(map),0);
18 for i:=1 to m do
19     for j:=1 to n do read(map[i,j]);
20 for i:=1 to m do begin
21     for j:=1 to n do begin
22         t:=slick(i,j); Inc(s,Ord(t>0)); Inc(p[t]);
23     end;
24 end;
25 Writeln(s);
26 for i:=1 to 10000 do begin
27     if p[i]>0 then Writeln(i, '',p[i]);
28 end;
29 end.
```

Література

- [1] **Ахо А., Хопкрофт Дж., Ульман Дж.** *Структуры данных и алгоритмы.* / Пер. с англ. – Москва: Вильямс, 2007. – 400 с.
- [2] **Беллман Р., Дрейфус С.** *Прикладные задачи динамического программирования.* – Москва: Наука, 1965. – 458 с.
- [3] **Величко В.Є., Рубан. М.М., Батунина В.П., Устінов С.Є.** *Олімпіадні задачі з інформатики: Розв'язання задач II етапу Всеукраїнської олімпіади з інформатики – 2007, 2008 рр.* – Слов'янськ, 2009. – 36 с.
- [4] **Величко В.Є., Рубан. М.М., Устінов С.Є., Пірус Є.М.** *Олімпіадні задачі з інформатики: Розв'язання задач II етапу Всеукраїнської олімпіади з інформатики – 2009 р.* – Слов'янськ, 2010. – 59 с.
- [5] **Грэхем Р., Кнут Д., Паташник О.** *Конкретная математика. Основы информатики.* М.: Мир; БИНОМ. Лаборатория знаний, 2006. – 703 с.
- [6] **Долинский М.С.** *Алгоритмизация и программирование на Turbo Pascal от простых до олимпиадных задач.* СПб.: Питер, 2005. – 237 с.
- [7] **Иванов Б.Н.** *Дискретная математика. Алгоритмы и программы.* М.: Лаборатория Базовых Знаний, 2001. – 288 с.
- [8] **Кирюхин В.М. Окулов С.М.** *Методика решения задач по информатике. Международные олимпиады.* – М.: БИНОМ. Лаборатория знаний, 2007. – 600 с.
- [9] **Кнут Д.** *Искусство программирования, в 3-х томах.* М.: Вильямс, 2000.
- [10] **Кормен Т., Лейзерсон Ч., Риверст Р., Штайн К.** *Алгоритмы: построение и анализ.* М.: Вильямс, 2005. – 1296 с.
- [11] **Левитин Ананий В.** *Алгоритмы: введение в разработку и анализ.* М.: Вильямс, 2006. – 576 с.
- [12] **Меньшиков Ф.** *Олимпиадные задачи по программированию.* – СПб.: Питер, 2005. – 320 с.
- [13] **Окулов С.М.** *Программирование в алгоритмах.* – М.: БИНОМ. Лаборатория знаний, 2004. – 341 с.
- [14] **Порублев И.Н., Ставровский А.Б.** *Алгоритмы и программы. Решение олимпиадных задач.* – Москва: Вильямс, 2007. – 480 с.
- [15] **Радион В.С.** *Олимпиады по информатике: задачи, решения, тесты.* – Минск: Аверсэв, 2007. – 367 с.

- [16] **Рубан. М.М., Устінов С.Є.** *Олімпіадні задачі з інформатики: Розв'язання задач II етапу Всеукраїнської олімпіади з інформатики – 2010 р.* – Слов'янськ, 2011. – 122 с.
- [17] **Седжвик Р.** *Фундаментальні алгоритми на С.* Київ: ДіаСофт, 2001. – 688 с.
- [18] **Скиена С.С., Ревилла М.А.** *Олімпіадні задачі по програмуванню. Руководство по подготовке к соревнованиям/ Пер. с англ.* – М: КУДИЦ-ОБРАЗ, 2005. – 416 с.
- [19] **Уорен Генри С.** *Алгоритмічні трюки для програмістів, исправленне издание / Пер. с англ.* – Москва: Вільямс, 2004. – 288 с.
- [20] **Шень А.** *Програмування: теореми і задачі.* – 2-е изд., испр. и доп. – М.: МЦНМО, 2004. – 296 с.

Ресурси мережі Internet

http://acmicpc.org	Офіційний сайт АСМ-ІСРС.
http://e-olimp.com	Система підготовки та проведення олімпіад зі спортивного програмування.
http://acm.lviv.ua	АСМ-Contester Український портал АСМ-спільноти.
http://acm.mipt.ru	Олімпіади з програмування на Фізтехі.
http://acm.timus.ru	Timus Online Judge — найкрупніший в Росії архів задач з різноманітних змагань зі спортивного програмування.
http://acmsolver.org	Портал для тих кого цікавлять змагання з програмування АСМ-ІСРС.
http://g6prog.narod.ru	Розбір олімпіадних задач з інформатики від Михайла Густокашина.
http://ips.ifmo.ru	Російська Інтернет-школа інформатики та програмування.
http://acmp.ru	Школа програміста. Красноярський край.
http://informatik.kz	Інформатика викладання та навчання.
http://www.olimp.sc170.kharkov.ua	Зимова школа з програмування, Харківський Національний Університет Радіоелектроніки.

Микола Миколайович Рубан
асистент кафедри алгебри
Слов'янський державний педагогічний університет

Сергій Євгенович Устінов
інженер-програміст лабораторії ОТІ
Слов'янський державний педагогічний університет

ОЛІМПІАДНІ ЗАДАЧІ З ІНФОРМАТИКИ

**Розв'язання задач II етапу
Всеукраїнської олімпіади з інформатики — 2011 р.**

ВИПУСК 11
СЕРІЯ: Викладачі СДПУ — учням, студентам, вчителям...

Відповідальний за випуск:

М.М. Рубан,
асистент кафедри алгебри
Слов'янський державний педагогічний університет

Підписано до друку 05.04.2012 р.
Формат 60x84 1/16. Ум. др. арк. 7,75.
Тираж 100 прим. Зам. № 327.

Підприємець Маторін Б.І.

84116, м. Слов'янськ, вул. Г.Батюка, 19.
Тел./факс +38 06262 3-20-99. E-mail: *matorinb@ukr.net*

Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру видавців,
виготівників і розповсюджувачів видавничої продукції ДК №3141, видане Державним
комітетом телебачення та радіомовлення України від 24.03.2008 р.
