

ПОСТФАКТУМ

(неопублікована стаття як штрих до обласного етапу учнівської олімпіади з програмування 2018 р.)

“Все минає”, “І це пройде”, та в кінцевому рахунку виявляється “Ніщо не проходить”. Ми поступово усвідомлюємо це з роками, повторюючи досвід царя Соломона. Початок учнівських олімпіад з програмування повертає в далекі 1987 (районні етапи) та 1988 (обласні етапи) роки. Олімпіади минали, поступово розвивалися, удосконалювалися, разом з тим старіли за змістом і формою, сьогодні, схоже, поступово перетворюються в анахронізми, тому потребують модернізації та осучаснення.

Останніми роками осучаснення відбувається за рахунок проведення в режимі онлайн безпосередньо в районах. Тут поряд з перевагами видимі й недоліки, адже ніхто не може забезпечити присутності на місцях членів оргкомітету, що може призвести і призводить до фальсифікацій. Крім того, олімпіади проводяться фактично у формі інтернет-олімпіад, оргкомітет звужується до однієї чи кількох персон, що в ці дні працюють з онлайн-сервером, вони можуть бути недостатньо кваліфікованими чи досвідченими, не виключено, що й частково недбалими. Компіляція, тестування розв’язків учасників і оприлюднення результатів непрозорі. Фактично сталося анулювання процедури апеляцій, адже в сучасних умовах апелюції можуть бути досягнуті тільки через перевірку розв’язків за допомогою тестів, які необхідно оприлюднювати для всіх учасників відразу після закінчення туру. Але тут можуть виникнути й додаткові труднощі, зокрема через використання у центрі та на місцях різних версій компіляторів мов програмування, різної потужності комп’ютерів тощо. А в результаті згаданих обставин порушується головний принцип олімпіади – рівність умов для всіх учасників, що фактично її робить формальною, а отже непотрібною. І з цим необхідно щось робити.

Щоб уникнути голослівності та недостатньої аргументованості змодельюємо ситуацію на прикладі III етапу олімпіади (візьмемо задачі *A*, *B* та *C* першого туру). Ці задачі вибрано через їх простоту, інколи майже тривіальність, що дозволяє спростити аргументацію, але ще й тому, що на цих задачах започатковано в олімпіаді з програмування 2018 введення додаткових умов щодо системи оцінювання, встановлено не зовсім вдалий до ситуації термін “підзадача”.

Задача A. РІВНЯННЯ Леді написала у свій електронний зошит з математики рівняння, що містить три цілих числа, знак рівності та одну з чотирьох основних арифметичних операцій (додавання, віднімання, множення та ділення). На жаль, вірус знищив знак рівності та операцій з ноутбука Леді. Допоможіть Леді відновити рівняння з трьох цілих чисел.

Формат вхідних даних: У першому рядку вхідного файлу знаходяться три цілих невід’ємних числа, менших за 100. Вхідні дані гарантують, що рішення завжди існує.

Формат вихідних даних: Виведіть знайдене рівняння, яке містить три цілих числа (в тому ж порядку), знак “=” і знак однієї з чотирьох арифметичних операцій. Якщо є декілька рішень, виведіть будь-яке з них.

Приклади:

<i>Equation.in</i>	<i>Equation.out</i>
5 3 8	5+3=8
5 15 3	5=15/3

Не будемо прискіпуватись до слів, термінів і т.д., залишивши відповідальність за формулювання на совісті авторів задач («рішення» замість «розв’язання», «рівняння» замість «рівність», «формат вихідних даних» замість «результати»). Але дивує, коли в умові задачі обласного етапу олімпіади, коли в преамбулі умови маються на увазі цілі числа, а в форматі вхідних даних замість них називаються цілі невід’ємні числа. Дивно також, що обмеження на них 100, а не, наприклад 10^9 . Про обмеження по часу, по пам’яті та на систему оцінювання згадувати для цієї задачі не будемо.

У запропонованому варіанті задача розв’язується надто просто:

```

program Z_A;
var a,b,c:real; f:text;
begin
Assign(f,'A8.dat');Reset(f);Read(f,a,b,c);
Assign(f,'A8.res');Rewrite(f);
if a+b=c then Write(f,a,'+',b,'=',c)
else if a-b=c then Write(f,a,'-',b,'=',c)
     else if a*b=c then Write(f,a,'*',b,'=',c)
          else if a/b=c then Write(f,a,'/',b,'=',c)
                else if a=b+c then Write(f,a,'=',b,'+',c)
                     else if a=b-c then Write(f,a,'=',b,'-',c)
                          else if a=b*c then Write(f,a,'=',b,'*',c)
                               else Write(f,a,'=',b,'/',c);

Close(f);
end.

```

Задача В. ЛАНЧ-БОКСИ

Обмеження по часу 0,1 секунди. Обмеження по пам’яті 64 Мб.

Леді працює менеджером ресторану. В ресторані приготували N ланч-боксів і Леді планує роздати їх в деякі школи. Припустимо, що є M шкіл та i -та школа замовляє k_i ланч-боксів. Леді намагається поширювати ланч-боксы для максимально можливої кількості шкіл. Крім того, Леді забобонна, а тому у неї є правило – для i -ої школи вона дає або нуль, або k_i ланч-боксів. Напишіть програму, яка допоможе Леді знайти максимальну кількість шкіл, які зможуть отримати ланч-боксы?

Формат вхідних даних: перший рядок вхідного файлу містить 2 цілих числа N та M . Далі йдуть M рядків: i -й рядок містить ціле число k_i .

Формат вихідних даних: виведіть у вихідний файл одне ціле число – максимальну кількість шкіл, які зможуть отримати ланч-боксы.

Приклад

<i>boxes.in</i>	<i>boxes.out</i>	<i>Пояснення</i>
10 4 3 9 4 2	3	Ланч-бокси зможуть отримати перша, третя та четверта школи

Система оцінювання: задача складається із трьох підзадач. Для підзадач виконуються додаткові обмеження, вказані в таблиці. Бали за підзадачу будуть нараховані лише за умови проходження усіх тестів підзадачі.

<i>Підзадачі</i>	<i>Бали</i>	<i>Обмеження</i>	<i>Примітка</i>
0	0		Тести з умови
1	20	$M = 1; 1 \leq N \leq 60000; 1 \leq k_i \leq 30000$	
2	30	$1 \leq M \leq 1000; 1 \leq N \leq 60000; 1 \leq k_i \leq 1000$	
3	50	$1 \leq M \leq 60000; 1 \leq N \leq 60000; 1 \leq k_i \leq 30000$	

Для початку кілька, можливо, несуттєвих зауважень, але які звертають увагу на недостатню продуманість вже на етапі підготовки умов задач та тестів:

- 1) Без “Леді” із її забобонністю ніяк не обійтись? Леді чи Степан (як було в умовах задач попередніх років) не має жодного відношення до змісту задачі, а лише захарашує умову, робить її важкочитабельною. Можна вважати, що так саме не має значення, як писати “ k_i ” чи “ k_i ”, але останнє все ж трохи спрощує читабельність. А “Леді” згадується п’ять та “ланч-бокси” аж шість разів!
- 2) Пригадується Всеукраїнський етап олімпіади багаторічної давності. Там була задача про міський автобус із 2,5 тисячами місць. Зрозуміло, таке число в тести вставлене для ускладнення розв’язку за рахунок обмежень, без яких, безперечно, не обійтись. Але чи не смішно, що шкіль може бути до 60000 (їх в усій Україні в кілька разів менше, а в столиці лише кількасот). А потужності якого ресторану дозволяють виготовляти за один день ланч-боксів 60000 комплектів? Це явно невдалі спроби внести компетентісний з креативнісним смаком підходи, чи бозна-що інше із серії інновацій, які сьогодні занадто модні і вживаються де треба, а де й ні.

В даній задачі не може йтись про власне підзадачі чи окремі варіанти розв’язку для кожної з них. Це всього акцент на варіантах тестів і обмеження, які накладаються в задачах.

Дійсно, навіть в авторському розборі цієї задачі сказано: “Нескладно помітити, що щоб максимізувати кількість шкіл, які зможуть отримувати ланч-бокси завжди вигідно виконувати замовлення шкіл, які замовили найменше ланч-боксів. Отже, відсортувавши масив k , роздаватимемо ланч-бокси школам у порядку зростання значення k , аж допоки не роздамо всі, або не роздамо всім школам”.

Але облишмо ці дійсно непринципові зауваження. В конкретному місці проведення етапу олімпіади при виконанні завдання і відсилки розв’язку на сервер виявилось, що так звана перша підзадача не проходить по тестах. Сервер

повідомив, що компіляція відбулась успішно, 19 із 20 тестів першої підзадачі зараховано, а останній ні, тому і всі 20 балів за першу підзадачу чи блок із двадцяти тестів виявився без зарахування балів. В той же час сервер повідомив, що деяким учасникам протягом чверті години зараховано за задачі *A* та *B* по 100 балів. Що б зробив досвідчений і зацікавлений учасник, якому перша задача була оцінена також на 100 балів, а в другій виявилась вищеописана ситуація? Розуміючи, що без цих початкових 20-и балів він не одержить і решти із ста, він продовжував би “копатися” у задачі *B*. У таких “копаннях” пройшло десь півтори години, дорогоцінний час, енергія і сили спливли, розв’язування решти трьох задач пройшло без ентузіазму, можливі навіть несправедливі нарікання на щось “нечисте”, як от: дехто раніше знав умови задач, комусь підсудили і т.д. (про непрозорість проведення олімпіад в режимі онлайн було вище). Другий тур наступного дня пройшов з тими ж результатами і таким же настроєм і успіхом.

Я неодноразово звертався по електронній пошті до організаторів олімпіади (персонально або опосередковано до Сергія Петрова (електронні адреси: serg_pet@sumdu.edu.ua та sergpet@gmail.com) із проханням надіслати тести по яких перевірялась задача *B* I туру III етапу олімпіади в Київській області, останні рази 4.05 та 14.05, але відповіді не одержав. Останні два рази у листі я надіслав також виділене тільки згідно умов першої підзадачі спрощене часткове розв’язання, яке гарантовано повинне давати всі 20 балів при будь-яких тестах із вказаного обмеження:

```
program B_I_Tur;
var N,M,k1:integer; f:text;
begin
Assign(f,'Boxes.in');ReSet(f);
Read(f,N,M,k1);Close(f);
Assign(f,'Boxes.out'); ReWrite(f);
if k1<=N then Write(f,1) else Write(f,0);
Close(f);
end.
```

В цьому розв’язку при наявності лише однієї школи не потрібні ні сортування, ні перебори, ні використання динамічних масивів, не вимагається вихід за межі цілочисельного типу *integer*, отже задача не може не вкластись у відведений час чи займе більше 64 Мб пам’яті.

А тепер наведемо програму для розв’язування тієї ж задачі *B* першого туру, яка враховує всі три так звані підзадачі із обмеженнями на вхідні дані:

```
program B;
var n,m,i,k,s:integer;
A:array[1..60000] of integer;
f:text;
procedure Inpt;
begin
Assign(f,'b9.in');ReSet(f);
ReadLn(f,n,m);
for i:=1 to m do
```

```

    ReadLn(f,A[i]);
    Close(f);
end;
procedure Sort(l,r:integer);
var i,j,x,y:integer;
begin
    i:=l;j:=r;
    x:=A[(l+r)div 2];
    repeat
        while (A[i]<x)do Inc(i);
        while (x<A[j])do Dec(j);
        if(i<=j)then begin
            y:=A[i];A[i]:=A[j];A[j]:=y;
            Inc(i); Dec(j)end;
        until(i>j);
        if(l<j)then Sort(l,j);
        if(i<r)then Sort(i,r);
    end;
procedure Proc;
begin
    Sort(1,m);i:=1;k:=0;
    while(i<=m)and(s<=n)do begin
        s:=s+a[i];k:=k+1;i:=i+1 end;
    if s>n then k:=k-1;
end;
procedure Outp;
begin
    Assign(f,'b9.out');ReWrite(f); Write(f,k);Close(f);
end;
begin
    Inpt; Proc; Outp;
end.

```

Ця програма має цілком прозорий з точки зору читабельності вигляд, працює в будь-якому середовищі програмування (*TurboPascal*, *BorlandPascal*, *FreePascal* чи іншому компіляторі *Pascal*, навіть в інтерпретаторі *PascalABC*), якщо зробити косметичні зміни при оголошенні цілочисельних величин (*integer*, *longint*, *int64* тощо), крім яких умова не передбачає інших типів. Єдина проблема, яка може виникнути, коли доведеться економити пам'ять, для чого слід застосовувати динамічні масиви. Але, як свідчить зовсім прозорий аналіз, при наявності ста тестів, допустивши, що хоч половина з них не потребують застосування динамічних масивів ця програма мусить бути оцінена хоча б половиною із ста балів, тим більше вона подолає першу підзадачу нам 20 балів, якби не накладені умови в так званих підзадачах.

Не бачивши тестів, якими оцінювалась задача на сервері, нами були складені власні, тут наведемо опис лише трьох із них (для економії місця розмістивши дані про заклади шкіл на ланч-бокси не вертикально, а горизонтально із деякими скороченнями).

Тест 1

60000 60000

1 1 1 ... 1 (всього 60000 чисел)

Тест 2

60000 60000

1 2 3 ... 59999 60000 (всього 60000 чисел)
--

Тест 3

60000 60000

100 1 2000 350 ... (всього 60000 не відсортованих чисел)
--

Перші два тести (вхідні дані та відповідні результати) скласти досить неважко, якщо скористатись послугами *Excel*. Розглянемо тест 2, в якому елементами лінійного масиву є члени арифметичної прогресії з першим елементом, та різницею, рівними по одиниці.

Змодельовавши таблицю, як показано на малюнку, із стовпчиком даних (із 60000 рядків) та стовпчиком відповідей, де праворуч від числа n запишемо суму n перших попередніх елементів з допомогою формули “ $=B(n-1)+An$ ”. З такої таблиці видно, що сума використаних згідно умови задачі ланч-боксів таким чином забезпечить 345 шкіл із сумарною кількістю 59685 ланч-боксів. Якщо вважати, що кожна із 60000 шкіл замовить лише по одному ланч-боксу, їх буде потрібно також 60000.

	A	B	C	D	E
335	335	56280			
336	336	56616			
337	337	56953			
338	338	57291			
339	339	57630			
340	340	57970			
341	341	58311			
342	342	58653			
343	343	58996			
344	344	59340			
345	345	59685			
346	346	60031			

Обидва тести абсурдні з точки зору реальності (шкіл в Україні менше 60000), вони не можуть замовляти лише по одному ланч-боксу (хіба що для одного директора). А якщо взяти вищезгадану арифметичну прогресію, то не треба десятків тисяч шкіл, досить всього 345, співрозмірно із м. Київ, але найбільше може замовити кожна школа лише 345 ланч-боксів (це невелика сільська школа).

Цікавіший приклад тесту за номером 3. Його легко змодельовати, вставивши між членами арифметичної прогресії якісь числа (хоч і однакове у всіх випадках, наприклад число 200). Та в такому випадку кількість шкіл також не буде більше 345, скоріше менше. Питається: для чого тоді такі непродумані обмеження?

Висновок однозначний: задача невдала. Хіба не можна зробити умову більш правдоподібною і реалістичною? При браку фантазії авторам задачі лише й залишалось, що “городити город” про підзадачі та пускаючи туман учасникам персонажами з дивними іменами (мається на увазі “Леді”).

Може здатись, що в останніх абзацах проглядається прихована іронія, недоречна в серйозній предметній статті, але як мимовільно не зіронізувати на кшталт “не смішіть мої капці”! Насправді ж тут цілком серйозно піднімається

питання модернізації підготовки та проведення олімпіад з програмування, значення яких для майбутнього дуже важливе, а сьогодні спостерігаються ознаки їх стагнації або угасання.

Найцікавіше бачимо в задачі С (також І туру).

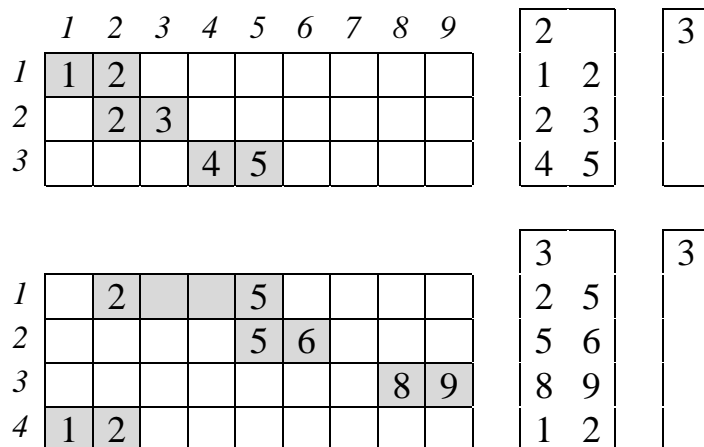
Задача С. МАГІЧНИЙ НАБІР¹ Набір називається магічним, якщо в ньому рівно N відрізків та кожна пара має спільну точку (точку чи точки?). Будемо вважати, що набір був магічним. Складено частковий магічний набір (містить частину відрізків, які утворюють магічний набір). Додано один відрізок, що не доповнює магічного набору (зліва чи справа). Після перемішування відрізки розміщені в довільному порядку. Доданий відрізок може бути тільки на зліва чи тільки справа.

Аналіз задачі: Спочатку потрібно навчитись знаходити перетин для двох відрізків. Нехай координати першого - (l_1, r_1) , другого (l_2, r_2) , причому $(l_1 < l_2)$. тоді їх перетин рахується наступним чином:

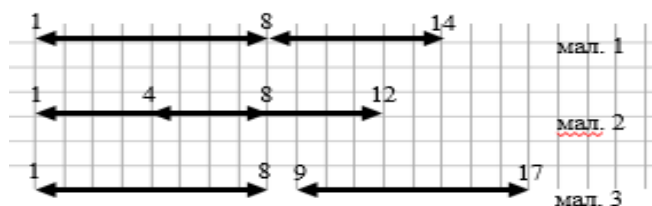
якщо $r_1 < l_2$, то їх перетин дорівнює 0;

якщо $r_1 \geq l_2$, то це відрізок $\max(l_1, l_2), \min(r_1, r_2)$

Давайте порахуємо перетин всіх відрізків на всіх префіксах та на всіх суфіксах². $pref[i]$ – перетин всіх підвідрізків від 1 до i , $suf[i]$ – перетин всіх підвідрізків від i до n , це можна порахувати за лінійне проходження по масиву. Тепер переберемо елемент, який могла додати Марічка, нехай його номер i , тоді потрібно перевірити що перетин між $pref[i - 1]$ і $suf[i + 1]$ дає непорожній перетин.



На малюнках 1-3 зображено три варіанти розміщення двох відрізків із



¹ Умову для зрозумілості і без спотворення змісту спрощено.

² Уточнити б, що тут розуміють під термінами «суфікси» та «префікси».

вказанням координат (l_i, r_i) . В умові не вказано, як розміщені відрізки, (мають одну спільну точку – мал.1, чи можуть мати в перерізі спільний відрізок – мал.2). Також не вказано, чи до обіду Леді склала один фрагмент магічного набору, чи два (мал.3). Залежно від цього сестра могла додати відрізок між відрізками Леді, тоді можливі чотири випадки, з яких лише один випадок дозволяє встановити вкладений сестрою відрізок. Якщо сестра додала відрізок, що лежить за межами, тобто справа або зліва від часткового магічного набору Леді, то можливі ще чотири варіанти, з них половина також не дає змоги визначити підкладений сестрою відрізок. Те ж непорозуміння виникає і при випадку, зображеному на мал.2. Висновок: задача С в поданому вигляді не може бути пропонованою учасникам олімпіади.

Тут проаналізовано лише три перші задачі I туру III етапу олімпіади. Обговорення решти семи задач, як і цієї статті, із зрозумілих причин опустимо, воно може бути проведене під час тематичного семінару обласного рівня. Але до решти задач і проведення олімпіади з програмування для учнів ЗНЗ в цілому кожен уважний вчитель знайде ще багато.

Як на мене, слід ретельніше добирати самі завдання, умови оцінювання та тести до них – це по-перше. А по-друге, слід обов'язково відновити повноцінні апеляції для учасників шляхом оприлюднення оригінальних тестів відразу після закриття етапу та його турів. Чи, можливо, прохання одержати ті злополучні вхідні дані не є обов'язком організаторів олімпіади, а особливо на вимогу?

Роками звучить слово гідність, але ж тут прозоро видно якраз пониження гідності учня. Звідки з'явиться ентузіазму? Слід також пам'ятати, що для учасників змагань, особливо початківців на ранніх етапах олімпіад, важлива участь, а не призи. Школа чекає забезпечення масовості та підняття зацікавленості програмуванням, яке на сучасному етапі дуже ослабло. Сьогодні бачиться протилежна картина – участь в районній олімпіаді з програмування бере активно чи пасивно лише не більше 3-5 шкіл району, в якому є не менше двадцяти загальноосвітніх шкіл. Спостерігається також, що за винятком учасників однієї-двох шкіл району, решта виконують роль вічних статистів.

Безперечно, все минає, минеться і це теж, але, при всьому, нічого не проходить безслідно. Ось такий гіркий постфактум, коли слідувати притчі про Соломона.

Вчитель інформатики Щасливського НВК Бориспільського району
В. Остапець