

КОМАНДНИЙ КУБОК УКРАЇНИ З ПРОГРАМУВАННЯ СЕРЕД УЧНІВСЬКОЇ МОЛОДІ

Присяжнюк А.В., Міланін О.А.

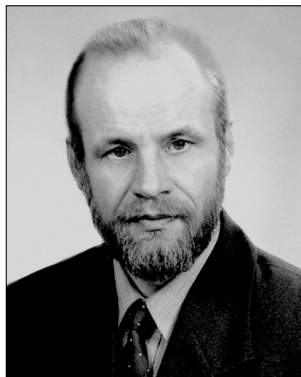
З метою пошуку, підтримки, розвитку творчого потенціалу обдарованої молоді, стимулювання творчого самовдосконалення учнів, зацікавлення їх у поглибленому вивченні інформатики; виявлення і розвиток обдарованих учнів, сприяння розвитку алгоритмічного мислення у школярів, підвищення інтересу до програмування; і відповідно до Положення про Всеукраїнські учнівські олімпіади з базових і спеціальних дисциплін, турніри, конкурси-захисти науково-дослідницьких робіт і конкурси фахової майстерності, на виконання наказу Міністерства освіти і науки України від 21.09.2010 р. №891 «Про проведення Всеукраїнських учнівських олімпіад і турнірів у 2010/2011 навчальному році, 02.04–16.04.2011 відбувся командний Кубок України з програмування серед учнівської молоді. Турнір проводився на базі Житомирського Державного Університету ім. Івана Франка за сприяння журналу АПН України «Комп'ютер у школі та сім'ї» і спонсорської допомоги ТОВ «ЗОП» Арніка» (м. Київ).

До участі в I етапі командного Кубка України 2011 серед школярів запрошувались усі бажані команди кожного загальноосвітнього навчального закладу України. Відбірковий етап проводився дистанційно на базі Інтернет-порталу організаційно-методичного забезпечення і дистанційних олімпіад з програмування для обдарованої молоді навчальних закладів України <http://e-olimp.com.ua>. За результатами I етапу змагань (I тур — 02.04.2011) 12 кращих команд отримали право участі у фінальному II етапі Кубка України серед школярів, який пройшов 16.04.2011 у Житомирському Державному Університеті.

Регламент Кубка був аналогічний регламенту студентської першості світу з програмування (<http://cm.baylor.edu/>):

- Команда складалася з трьох учнів з одного навчального закладу.
- Кожній команді надавався один комп'ютер.
- Командам було запропоновано 10–15 задач різної складності, викладених українською, російською й англійською мовами, тривалість змагань — 5 годин без перерви.
- Користування допоміжними джерелами інформації, словниками й електронними пристроями не дозволялося.
- За дотриманням описаних правил на I етапі відповідав тренер команди, або регіональний представник журі, на II етапі — журі.
- Мовами програмування є: C++, Java, Pascal.
- Перевірка розв'язків здійснювалась автоматично, через Інтернет, з використанням он-лайн системи проведення олімпіад <http://e-olimp.com.ua>.

Під час відбору на фінальну (очну) частину Кубка (II етап) діяло обмеження — від одного навчального закладу на фінальну частину могло бути допущено не більше 3-х команд, які посіли, відповідно, вищі місця. За результатами турніру командний Кубок України 2011 серед школярів виборола команда ЛНВК



«Школа першого ступеня — гімназія» (м. Львів) у складі Макар Андрій та Макар Тарас.

Нижче наведені задачі фінальної частини турніру й ідеї щодо їх розв'язання.

Залізниця (О. Міланін)

Як це не дивно, але у нас в країні прийнято спочатку робити, а потім думати. Так і цього разу: Мінтранс видав наказ створити у країні залізничну мережу, і тут же буквально за місяць вся країна була покрита рейками. Звичайно, вирішили зекономити і не стали проводити більше однієї колії між двома містами. І ось вже все було готово для того, щоб пустити перший потяг, як у справу вступили СБ і МНС. Вони ж постановили, що з метою безпеки по кожному шляху поїзди будуть ходити завжди у одному напрямку, і що в наших інтересах якомога швидше визначити, у якому. Отже, постала серйозна проблема — призначити кожному залізничному шляху, який з'єднує безпосередньо два міста, один з двох можливих напрямків. При цьому у Мінтрансі заважали так призначити напрямки, щоб можна було пустити якомога більше потягів у цій новій системі залізничних доріг. Між двома містами можна пустити не більше одного потягу і лише у тому випадку, якщо з одного міста можна потрапити в інше і повернутись назад з врахуванням напрямків залізничних шляхів.

Вам задано план залізничних шляхів. Потрібно вивести лише максимальну кількість потягів, які можна пустити за оптимального вибору напрямків. Напрямки визначати не потрібно — все-таки у міністерстві транспорту працюють гідні люди — вони це зроблять самі.

Вхідні дані. У першому рядку дано два числа N і K ($1 \leq N \leq 10000$, $0 \leq K \leq 100000$) — кількість міст і збудованих залізничних шляхів відповідно. У наступних K рядках по два числа A і B ($1 \leq A, B \leq N$) — номери міст, між якими було збудовано пряму дорогу.

Вихідні дані. У єдиному рядку вивести максимальну кількість запущених потягів.

Приклад вхідних даних	Приклад вихідних даних
6 7 1 2 2 3 3 1 4 5 5 6 6 4 3 4	6

Ідея розв'язання задачі

У задачі було задано неорієнтований граф, потрібно було орієнтувати ребра так, щоб кількість таких пар вершин $\{u, v\}$, що з u існує шлях у v і навпаки, була максимальною.

Розкладемо заданий граф на двозв'язні компоненти. Ті вершини, які потрапили у різні компоненти, при довільній орієнтації ребер не зможуть утворити пару. Так як, або вони взагалі не зв'язні, або на довільному шляху з однієї в іншу, є міст. Цей міст можна орієнтувати лише в одну сторону, тобто обидва орієнтовних шляхи одночасно існувати не можуть. З іншого боку, усі вершини, які потрапили в одну двозв'язну компоненту, можна при правильній орієнтації ребер перетворити у сильнозв'язну компоненту. Тоді всі вершини будуть формувати пари. Із двозв'язної компоненти можна конструктивно створити сильнозв'язну, якщо орієнтувати ребра у процесі обходу у глибину. А саме, у якому напрямку вперше розглянуто ребро, у такому й орієнтувати.

Тоді відповіддю буде

$$\sum_i \frac{n_i \cdot (n_i - 1)}{2},$$

де n_i — кількість вершин у i -й двозв'язній компоненті.

Юстас—Алексу (О. Міланін)

Після блискуче проведеної операції Штірліц зміг визначити чисельність фашистської армії. Звичайно таку інформацію вже чотири роки як чекають у штабі радянської армії. Щоб спілкуватись зі штабом Штірліц використовує n радистів. Кожен із радистів повинен передати повідомлення від Штірліца у штаб. Штірліц, як хитрий розвідник, зашифрував своє послання так: кожному з радистів він дав одне і те ж число — чисельність армії, але у своїй системі числення, та ще й так, що всі основи систем числення у радистів попарно взаємно прості. Після передачі радіограми нішпорки Мюллера змогли визначити останній символ кожного з повідомлень. Ви працюєте штатним програмістом і повинні визначити, яке мінімальне число міг відправити Штірліц у своєму повідомленні. Мюллер на відміну від вас не дуже любить бінарний код, тому він хоче, щоб шукане число ви вивели у десятковій системі числення.

Вхідні дані. У першому рядку задається число n — кількість радистів у Штірліца. У наступному рядку знаходяться n чисел a_i — основи систем числення, у яких Штірліц давав повідомлення радистам ($2 \leq a_i \leq 36$). У третьому рядку через проміжок записано n символів c_i — останні літери кожного з повідомлень ($0 \leq c_i < a_i$; c_i — або цифра від 0 до 9, або літера від A до Z).

Вихідні дані. Вивести мінімальне число, яке міг передати Штірліц у десятковій системі числення.

Приклад вхідних даних	Приклад вихідних даних
2 6 13 1 B	37

Ідея розв'язання задачі

У задачі потрібно було знайти таке мінімальне число x , для якого виконуються рівності $x \bmod a_i = c_i$. За умови, що всі основи систем числень

попарно взаємно прості, існує єдине число x по модулю Pa_i , яке має вказані вище властивості (китайська теорема про залишки).

Нехай у нас є пара обмежень $x \bmod a_1 = c_1$ і $x \bmod a_2 = c_2$, тоді $x = ka_1 + c_1$, усі операції цілочисельні. Тоді $(ka_1 + c_1) \bmod a_2 = c_2$, $k = a_1^{-1}(c_2 - c_1) \bmod a_2$. Так як a_1 і a_2 взаємно прості, то $a_1^{-1} \bmod a_2$ завжди існує і може бути знайдено за допомогою розширеного алгоритму Евкліда. Отже, ми задачу з двома обмеженнями звели до задачі з одним $x \bmod a = c$, де, $a = a_1 a_2$ а $c = ((a_1^{-1}(c_2 - c_1) \bmod a_2) a_1 + c_1) \bmod a$.

Отже, всі обмеження по черзі згортаються, доки не залишиться одне. Відповіддю є результуюче число.

Заготівля грибів (А. Присяжнюк)

Білочка восени готується до майбутньої зими і зайнята заготівлею грибів. Збирає вона лише білі гриби і підберезовики, потім нанизує їх на в'язку і вішає на дереві для просушування. При цьому Білочка хоче, щоб на кожній в'язці було рівно N грибів, усі в'язки були різними і при цьому щоб у довільній в'язці не зустрічалась послідовність «білий гриб» — «підберезовик» — «білий гриб». Скільки в'язок грибів на майбутню зиму заготовить Білочка, якщо у лісі цієї осені грибів достатньо багато.

Вхідні дані. Єдине натуральне число N ($N < 10000$) — кількість грибів у в'язці.

Вихідні дані. Кількість в'язок, які будуть заготовлені Білочкою на зиму. Так як відповідь може бути достатньо великою, відповідь вивести за модулем 9997.

Приклад вхідних даних	Приклад вихідних даних
Sample 1 3	Sample 1 7
Sample 2 4	Sample 2 12

Ідея розв'язання задачі

Сформулюємо задачу в такому формулюванні: скільки існує послідовностей довжини n , які складаються з 0 та 1 таких, що у них не зустрічається підряд блок 010. Задача розв'язується за допомогою динамічного програмування. $d(i, j, k)$ — кількість послідовностей довжини i таких, що останній елемент k , а передостанній j , $i \in [1, n]$, $j, k \in \{0, 1\}$. Тоді застосуємо динаміку $d(i, j, k) = d(i-1, 0, j) + d(i-1, 1, j)$, крім випадку $d(i, 1, 0) = d(i-1, 1, 1)$. Ініціалізується динаміка значеннями $d(1, 1, 0) = d(1, 1, 1) = 1$, $d(1, 0, 0) = d(1, 0, 1) = 0$. Відповідь до задачі підраховується як сума $d(n, j, k)$ по всіх j і k .

Автобани (А. Присяжнюк)

Мінтранс зробив відповідні висновки зі свого залізничного експерименту і вирішив перейти до будівництва автобанів, так як це добре вписувалось у плани підготовки до Євро-2012. Було вирішено між містами побудувати мережу автобанів, причому з метою економії коштів ці автобани повинні бути строго прямими відрізками. Звичайно, що і швидкість пересування і час проїзду на таких автобанах буде рівнем вище. Крім того, було прийнято рішення, що так як інколи якийсь автобан може закрива-

тись на реконструкцію, побудована мережа доріг повинна забезпечувати можливість доїхати від одного міста до іншого навіть під час проведення подібної реконструкції. Також зрозуміло, що кількість таких доріг у мережі повинна бути мінімальною.

Як завжди, знайшлась ще одна розумна голова, яка запропонувала плату за проїзд по автобану брати не при в'їзді на нього, а лише на перехрестях. Ось тепер і ламають голови у Мінтрансі: як побудувати подібну мережу з мінімальною кількістю доріг, але з максимальною кількістю пунктів збору коштів за проїзд?

Ваша задача — написати програму, яка підрахує максимальну кількість пунктів оплати проїзду по автобанах, які приносять прибуток, а як будувати саму мережу — вже будуть проектувати працівники Мінтрансу. Та, і ще — вже прийнято нові правила дорожнього руху для автобанів, які забороняють у пунктах оплати за проїзд з'їжджати з одного автобану і продовжувати рух по іншому — не будемо пояснювати, виходячи з яких економічних міркувань це було зроблено:

Вхідні дані. У першому рядку єдине натуральне число — кількість тестових випадків T ($T \leq 1000$). У наступних T рядках також задано по одному натуральному числу N — кількість міст у мережі ($N \leq 32767$).

Вихідні дані. Для кожного тестового випадку в окремому рядку вивести шукану максимальну кількість пунктів збору грошей.

Приклад вхідних даних	Приклад вихідних даних
3	0
3	1
4	5
5	

Ідея розв'язання задачі

Задачу можна математично переформулювати так: визначити максимальну кількість перетинів діагоналей, яку можна досягти у n -кутнику. Якщо $n \leq 3$, то відповідь 0, інакше, якщо n парне, то

$$\frac{n \cdot (n - 4)}{2} + 1, \text{ якщо ж непарне, то } \frac{n \cdot (n - 3)}{2}.$$

Рисове поле (М. Рубан)

Одного разу, подорожуючи по казковій країні, Кролик у одній з долин побачив рисове поле. Поле займало всю долину, яка є прямокутником розміром $M \times N$ клітин. Кролик був дуже вражений складною будовою цього поля. Воно складалося з окремих ділянок, кожна ділянка розташована на своєму рівні. Оскільки розміри поля дуже великі, то Кролик просить Вас допомогти йому знайти ділянки поля з найбільшою і найменшою площами.

Вхідні дані. У першому рядку вхідного файлу два числа M і N ($1 \leq M, N \leq 100$). У наступних M рядках по N чисел у кожному, розділених пропусками, йде опис карти. Кожна клітина карти містить цілі числа p_i ($0 \leq p_i < 100$), які вказують, на якому рівні перебуває ділянка. Ділянка — це набір клітин поля, які розташовані на одному рівні і мають з іншими клітинами цього ж рівня спільну сторону.

Вихідні дані. Вивести через пропуск два числа: максимальну і мінімальну площі ділянок.

Приклад вхідних даних	Приклад вихідних даних
77 7777777 7555557 7424457 7420457 7422257 7444457 7777777	24 1

Ідея розв'язання задачі

Задача розв'язується пошуком у глибину.

Дробі (А. Присяжнюк)

Для заданого натурального N вивести у порядку зростання всі правильні нескоротні дроби, знаменник яких не перевищує N .

Вхідні дані. У першому рядку задано натуральне число T — кількість тестових випадків ($T \leq 10$). У наступних T рядках задано по одному натуральному числу N ($1 < N \leq 2000$).

Вихідні дані. Для кожного тестового випадку вивести в порядку зростання всі правильні нескоротні дроби. Сусідні дроби повинні бути відокремлені комою і одним пропуском.

Приклад вхідних даних	Приклад вихідних даних
3	1/2
2	1/5, 1/4, 1/3, 2/5, 1/2, 3/5,
5	2/3, 3/4, 4/5
3	1/3, 1/2, 2/3

Ідея розв'язання задачі

Розв'язок полягає в тому, щоб згенерувати всі правильні нескоротні дроби, відсортувати їх і на запити виводити лише ті, знаменник яких не перевищує задане число. Детальніше: перебираємо чисельники й знаменники, якщо вони взаємно прості, то відповідний дріб додаємо до списку. Сортуємо список один раз швидким сортуванням, після чого опрацюємо всі запити.

Прості числа Маші (А. Присяжнюк)

Коли Маша у школі дізналась про прості числа і повідомила про це Миші, той їй по секрету повідомив, що практично довільне парне число можна подати у вигляді суми двох простих. Маша тут же почала перевіряти гіпотезу Миші, але при цьому вона хоче:

- щоб сума доданків була рівна заданому парному числу;
- щоб обидва доданки були простими;
- щоб різниця між доданками була мінімальною.

Напишіть програму, яка допоможе Миші перевірити вірність математичних досліджень Маші.

Вхідні дані. У єдиному рядку через пропуск перераховано послідовність парних чисел N ($6 \leq N \leq 20000$), які перевіряються Машею. Кількість чисел у одному прикладі не перевищує 100.

Вихідні дані. Для кожного тестового прикладу в окремому рядку виведіть шукану мінімальну пару простих чисел. Виводьте у парах числа, відсортованими за не спаданням.

Приклад вхідних даних	Приклад вихідних даних
20 30 40	7 13 13 17 17 23

Ідея розв'язання задачі

Задача розв'язується повним перебором. Для заданого числа n перебираємо всі i такі, що $2 \leq i \leq n/2$,

вибираємо таке i , що i та $i-n$ — прості числа і при цьому значення $n-2i$ мінімальне.

Маша і паліндром (А. Присяжнюк)

Маші у школі розповіли, що таке паліндром. І ось, прийшовши у гості до Миші, вона дістала його кубики, на яких зображені маленькі літери англійського алфавіту і почала грати сама із собою у гру, придуману по дорозі. Суть гри полягала у тому, що вона навмання складала слово — «абракадабру» із довільного числа кубиків, а потім пробувала отримати з цього слова паліндром.

Так як Маша ще маленька, то вона завбачливо у правилах ходів вказала, що під час гри можна міняти місцями лише два сусідніх кубики. Звичайно, інколи їй це вдавалось, а інколи — ні, і тоді вона здивовано вимовляла «Ooops...».

Ваша задача, знайти мінімальну кількість перестановок кубиків, потрібну Маші для отримання чергового паліндрому.

Вхідні дані. У першому рядку задано натуральне число — кількість тестових випадків T ($T \leq 150$). У наступних T рядках задана чергова початкова Машина послідовність з кубиків. Довжина послідовності не перевищує 100.

Вихідні дані. Вивести шукану мінімальну кількість ходів, потрібну Маші для отримання паліндрому, або, у випадку неможливості це зробити, здивовану фразу Маші «Ooops...» (без лапок).

Приклад вхідних даних	Приклад вихідних даних
3 mamad asflkj aabb	3 Ooops... 2

Ідея розв’язання задачі

Задано рядок, потрібно визначити, за яку мінімальну кількість обмінів сусідніх літер можна отримати паліндром. Якщо у рядку є хоча б дві літери, які зустрічаються непарну кількість разів, то паліндром отримати не можна. Якщо паліндром скласти можна, то оптимальний розв’язок буде складати його жадібно. Йдемо зліва праворуч по рядку, доки не дійдемо до середини. Нехай ми знаходимось у позиції i , з літерою s_i , тоді на позицію $n-1-i$ потрібно поставити таку ж літеру. Тому серед усіх літер на позиціях з $i+1$ по $n-1-i$ виберемо найближчу до $n-1-i$ рівну s_i і ланцюжком транспозицій поставимо на позицію $n-1-i$. Якщо такої немає, то поміняємо місцями літери на позиціях i та $i+1$. Отже, ми явно побудували паліндром, а відповіддю буде кількість транспозицій, які ми при цьому зробили.

Наступна перестановка (А. Присяжнюк)

У цій задачі ви повинні написати програму, яка приймає (можливо довго) рядок з десяткових цифр і яка повинна вивести перестановку цих десяткових цифр, що дає наступне за величиною за заданим десятковим число. Наприклад:

123 -> 132
279134399742 -> 279134423799

Цілком можливо, що вхідні дані можуть містити набір цифр, який не має шуканої наступної перестановки. Наприклад, 987.

Вхідні дані. Перший рядок вхідних даних містить єдине ціле число P , ($1 \leq P \leq 1000$), яке вказує на кількість тестових випадків. У кожному з наступних рядків міститься один тестовий випадок, де спочатку вказано номер тестового випадку, а потім через пропуск задано відповідний набір вхідних цифр, який складається з не більше ніж 80 десяткових цифр.

Вихідні дані. Для кожного тестового випадку відповідь виводьте в окремому рядку. Якщо для заданого набору цифр не існує наступної перестановки, вивести спочатку номер тестового випадку і далі через пропуск рядок BIGGEST. Якщо ж розв’язок існує, то виведіть спочатку також номер тестового випадку, а потім через пропуск знайдену наступну перестановку вхідних цифр.

Приклад вхідних даних	Приклад вихідних даних
3 1 123 2 279134399742 3 987	1 132 2 279134423799 3 BIGGEST

Ідея розв’язання задачі

Алгоритмічний розв’язок полягає в: нехай $p=p_1, \dots, p_n$ — перестановка, знайдемо таке максимальне i , що $p_i < p_{i+1}$. Якщо такого немає, то задана перестановка максимальна. У протилежному випадку виберемо $j \in [i+1, n]$ таке, що p_i — мінімально можливе з усіх $p_j > p_i$. Міняємо місцями елементи i та j , а всі елементи, починаючи з $i+1$, сортуємо за зростанням. Отримана перестановка і є відповіддю.

Найбільша зростаюча підпослідовність (М. Медведєв)

Розглянемо послідовність x_i , задану такими співвідношеннями:

$$x_0 = a + b, x_1 = a - b, \\ x_i = (a \cdot x_{i-2} + b \cdot x_{i-1}) \bmod m, i > 1$$

Для заданого натурального n знайти довжину найбільшої зростаючої підпослідовності $x_0, x_1, x_2, \dots, x_n$.

Вхідні дані. Кожен тест складається з одного рядка, який містить чотири натуральних числа a, b, m, n ($a=b, 1=a, b, m, n=10^6$). Кількість тестових випадків у одному тесті не перевищує 20.

Вихідні дані. Для кожного тесту в окремому рядку вивести довжину найбільшої зростаючої підпослідовності.

Приклад вхідних даних	Приклад вихідних даних
3 1 20 10 5 2 1000 2000	5 70

Ідея розв’язання задачі

Задано послідовність цілих чисел, знайти максимальну зростаючу підпослідовність. Нехай $d(i)$ — мінімальне число, на яке може закінчуватись зростаюча підпослідовність довжини i . Здійснимо прохід по заданій послідовності, підтримуючи масив d . Коли ми розглядаємо черговий елемент послідовності, потрібно знайти таке максимальне i , для якого $d(i)$ менше цього елемента. Тоді у $d(i+1)$ можна записати значення нашого елемента. Очевидно, що всі елементи у масиві d у довільний момент йдуть за зростанням, тому для знаходження i застосуємо бінарний пошук. Масив d потрібно ініціалізувати нескінченностями. Тоді відповіддю буде індекс останнього елемента в d , який менший нескінченності.