

Algorithm 2021, Qualification

🕒

28 ноя 2021, 17:40:33

🏁

старт: 3 окт 2021, 14:12:29

🏁

финиш: 3 окт 2021, 16:12:29

⌚

длительность: 02:00:00

🕒

начало: 27 сен 2021, 12:00:00

🕒

конец: 3 окт 2021, 23:59:00

А. НольОдин

Ограничение времени	1 секунда
Ограничение памяти	1Gb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Полное решение будет оценено в 2 балла.

Сравните два числа в двоичной системе счисления. Числа представлены последовательностью слов без пробелов, обозначающих цифры (0 — zero, 1 — one).

Формат ввода

Первая строка содержит запись первого числа s_1 ($3 \leq |s_1| \leq 1000$).

Вторая строка содержит запись второго числа s_2 ($3 \leq |s_2| \leq 1000$).

Числа не содержат лидирующих нулей.

Формат вывода

Выведите символ > (ASCII 62), если первое число больше второго, символ < (ASCII 60), если второе число больше первого, иначе выведите символ = (ASCII 61) .

Пример 1

Ввод 📄	Вывод 📄
oneone onezerozero	<

Пример 2

Ввод 📄	Вывод 📄
zero zero	=

Пример 3

Ввод 📄	Вывод 📄
onezero oneone	<

Пример 4

Ввод 📄	Вывод 📄
oneonezerozero onezeroonezero	>

Пример 5

Ввод 📄	Вывод 📄
one zero	>

Пример 6

Ввод 📄	Вывод 📄
one one	=

Язык

GNU c++17 7.3

Набрать здесь

Отправить файл

```
1 #include <bits/stdc++.h>
2 #define _CRT_DISABLE_PERFCRIT_LOCKS
3
4 using namespace std;
5
6 int main(int argc, char const *argv[]) {
7     ios_base::sync_with_stdio(false);
8     cin.tie(NULL);
9     cout.tie(NULL);
10    std::string s;
11    std::cin >> s;
12    std::string t1, t2;
13    for (int i = 0; i < s.length(); i) {
14        t1 += '0' + !s[i] == '1';
15        i += 3 + (s[i] == '1');
16    }
17    std::cin >> s;
18    for (int i = 0; i < s.length(); i) {
19        t2 += '0' + !s[i] == '1';
20        i += 3 + (s[i] == '1');
21    }
22    if (t1.length() != t2.length())
23        std::cout << (t1.length() < t2.length() ? '<' : '>') << '\n';
24    else {
25        for (int i = 0; i < t1.length(); i++) {
26            if (t1[i] != t2[i]) {
27                std::cout << (t1[i] < t2[i] ? '<' : '>') << '\n';
28                return 0;
29            }
30        }
31        std::cout << '=' << '\n';
32    }
33 }
34
```

Отправить

Следующая

Algorithm 2021, Qualification

🕒 28 ноя 2021, 17:46:26
старт: 3 окт 2021, 14:12:29
финиш: 3 окт 2021, 16:12:29
длительность: 02:00:00
начало: 27 сен 2021, 12:00:00
конец: 3 окт 2021, 23:59:00

В. Плитки 2x2

Ограничение времени	2 секунды
Ограничение памяти	1Gb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Решение, корректно работающее с картиной и плитками, покрашенными в белый и черный цвета, будет оценено в 2 балла.

Полное решение, корректно работающее с картиной и плитками, покрашенными в белый, черный и красный цвета, будет оценено в 3 балла (включая 2 балла за подзадачу выше).

Дан набор плиток 2×2 . У каждой плитки четыре квадрата покрашены в один из трех цветов: белый (W), черный (B) или красный (R).

Определите, можно ли из набора плиток составить пиксельную картину $n \times m$ (стороны картины имеют четный размер), если плитки при выкладывании не должны перекрываться и не могут выходить за пределы картины. При выкладывании плитки можно поворачивать, но нельзя ломать.

Плитка непрозрачная, и краска нанесена только с одной стороны плитки.

Формат ввода

В первой строке дано число k ($1 \leq k \leq 10^5$) — количество плиток в наборе.

В следующих $2k$ строках даны описания плиток. Описание каждой плитки занимает две строки по два символа, они задают цвета квадратов плитки.

Далее даны два числа n и m ($2 \leq n, m \leq 512$, n и m четные) — размеры картины. В каждой из следующих n строк дано описание очередного ряда картины.

При описании плиток и пикселей картины используются только символы W, B и R.

Формат вывода

В единственной строке выведите Yes, если из набора плиток можно собрать картину, иначе выведите No.

Пример 1

Ввод

Вывод

Ввод

Вывод

1
WW
BW
2 2
WB
WW

Yes

Пример 2

Ввод

Вывод

2
WW
BB
WB
WB
2 4
WBWB
WBWB

Yes

Пример 3

Ввод

Вывод

3
WW
WW
WW
WW
BB
BB
4 2
WW
BB
BB
WW

No

Пример 4

Ввод

Вывод

Ввод

Вывод

4
WW
WB
WW
WB
WW
WB
WW
WB
2 2
BB
BB

No

Пример 5

Ввод

Вывод

2
BW
WB
BW
WB
2 2
WW
BB

No

Пример 6

Ввод

Вывод

4
RR
RR
WW
WW
BB
BB
WW
WW
4 4
WWBB
WWBB
RRWW
RRWW

Yes

Пример 7

Ввод

Вывод

Ввод

Вывод

1
RW
RB
2 2
RB
RW

No

Примечания

В первом и втором примерах необходимо повернуть имеющиеся плитки.

В третьем примере можно было бы собрать картину, если бы было разрешено выкладывать плитки поверх размещенных.

В четвертом примере можно было бы собрать картину, если бы было разрешено ломать плитки или выходить за пределы картины.

Язык GNU c++17 7.3

Набрать здесь

Отправить файл

```
1 #include <bits/stdc++.h>
2 #define _CRT_DISABLE_PERFCRIT_LOCKS
3
4 using namespace std;
5
6 std::string unify(std::string& s) {
7     std::set<std::string> uni;
8     uni.insert(s);
9     for (int i = 0; i < 3; i++) {
10         s += s[0];
11         s.erase(0, 1);
12         uni.insert(s);
13     }
14     return *uni.begin();
15 }
16
17 int main(int argc, char const *argv[]) {
18     ios_base::sync_with_stdio(false);
19     cin.tie(NULL);
20     cout.tie(NULL);
21     int k;
22     std::cin >> k;
23     std::multiset<std::string> bars;
24     for (int i = 0; i < k; i++) {
25         std::string s, l;
26         std::cin >> s >> l;
27         s += l[l];
28         s += l[0];
29         bars.insert(unify(s));
30     }
31     int n, m;
32     std::cin >> n >> m;
33     std::vector<std::string> pic(n);
34     for (int i = 0; i < n; i++) {
35         std::cin >> pic[i];
36     }
37     for (int i = 0; i < n; i += 2) {
38         for (int j = 0; j < m; j += 2) {
```

Отправить

Предыдущая

Следующая

Algorithm 2021, Qualification

🕒 28 ноя 2021, 17:47:49
старт: 3 окт 2021, 14:12:29
финиш: 3 окт 2021, 16:12:29
длительность: 02:00:00
начало: 27 сен 2021, 12:00:00
конец: 3 окт 2021, 23:59:00

С. Шары и коробки

Ограничение времени	2 секунды
Ограничение памяти	1Gb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Полное решение будет оценено в 4 балла.

Есть цветные шары k цветов (a_i шариков цвета i). Нужно разложить все шары в коробки с выполнением следующих условий:

- во всех коробках должно совпадать суммарное количество шариков;
- в каждой коробке шариков цвета i должно быть не менее b_i ($0 \leq b_i \leq a_i$).

Максимизируйте количество коробок, в которые будут разложены шары.

Формат ввода

Первая строка входных данных содержит число k ($1 \leq k \leq 10^5$) — число цветов.

Во второй строке дано k чисел a_i ($1 \leq a_i \leq 10^5$, $\sum_{i=1}^k a_i \leq 10^5$) — количество шариков цвета i .

В третьей строке дано k чисел b_i ($0 \leq b_i \leq a_i$) — нижнее ограничение на количество шариков в каждой коробке.

Формат вывода

В первой строке выведите два числа: n — количество коробок, которые можно заполнить, и m — количество шариков в каждой коробке.

Далее выведите n строк по m чисел в каждой — цвета шариков, которые лежат в очередной коробке.

Суммарное количество шариков цвета i во всех коробках должно быть равно a_i .

Пример 1

Ввод	Вывод
5 1 2 3 4 5 1 2 3 4 5	1 15 1 2 2 3 3 4 4 4 4 5 5 5 5 5

Пример 2

Ввод	Вывод
1 10 0	10 1 1 1 1 1 1 1 1 1 1 1 1

Пример 3

Ввод	Вывод
5 4 5 5 5 5 1 1 1 1 1	4 6 1 2 3 4 5 5 1 2 3 4 5 4 1 2 3 4 5 3 1 2 3 4 5 2

Примечания

В первом примере все шары первого цвета должны находиться в одной коробке.

Во втором примере любое количество шариков может быть в одной коробке, поэтому можно положить в каждую коробку по одному шару.

В третьем примере нельзя использовать более четырех коробок, так как шариков первого цвета всего четыре.

Язык GNU c++17 7.3

Набрать здесь

Отправить файл

Algorithm 2021, Qualification

🕒 28 ноя 2021, 17:48:23
старт: 3 окт 2021, 14:12:29
финиш: 3 окт 2021, 16:12:29
длительность: 02:00:00
начало: 27 сен 2021, 12:00:00
конец: 3 окт 2021, 23:59:00

D. Матрица

Ограничение времени	1 секунда
Ограничение памяти	1Gb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Решение, корректно работающее в ограничениях $1 \leq nm \leq 2^4$, будет оценено в 2 балла.
Полное решение будет оценено в 4 балла (включая 2 балла за подзадачу выше).
Дана матрица $n \times m$ (n, m — степени двойки), заполненная целыми числами от 1 до nm (числа по возрастанию по строкам).
Например, для $n = 4$ и $m = 8$ матрица имеет следующий вид:

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32

За один шаг мы «складываем» матрицу пополам, как лист бумаги, поперек большей стороны (по горизонтали или по вертикали) и суммируем числа, которые накладываются друг на друга, до тех пор, пока не останется один элемент. Квадрат складываем по горизонтальной линии.
Элементы всех полученных матриц (в том числе и исходной) выписываем в одну последовательность.
Найдите количество различных выписанных чисел.

Формат ввода

В единственной строке входных данных записаны два числа n и m ($1 \leq n \times m \leq 2^{30}$). Оба числа являются степенями двойки.

Формат вывода

Выведите единственное число — количество различных выписанных чисел.

Пример 1

Ввод	Вывод
1 1	1

Пример 2

Ввод	Вывод
1 8	11

Пример 3

Ввод	Вывод
4 4	21

Язык GNU c++17 7.3

Набрать здесь Отправить файл

1

Отправить

Предыдущая

Следующая

Algorithm 2021, Qualification

⌚ 28 ноя 2021, 17:48:39

старт: 3 окт 2021, 14:12:29

финиш: 3 окт 2021, 16:12:29

длительность: 02:00:00

начало: 27 сен 2021, 12:00:00

конец: 3 окт 2021, 23:59:00

Е. Сортировка матрицы

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	2 секунды	1Gb	стандартный ввод или input.txt	стандартный вывод или output.txt
Python 3.7.3	7 секунд	1Gb		
Python 3.7 (PyPy 7.3.3)	7 секунд	1Gb		
C# (MS .Net 5.0)+ASP	3 секунды	1Gb		
Kotlin 1.4.30 (JRE 11)	3 секунды	1Gb		
OpenJDK Java 11	3 секунды	1Gb		
Golang 1.16	3 секунды	1Gb		

Решение, корректно работающее в ограничениях $1 \leq n, m \leq 10$, будет оценено в 2 балла.

Полное решение будет оценено в 6 баллов (включая 2 балла за подзадачу выше).

Дана матрица $n \times m$, состоящая из нулей и единиц. За один ход можно выбрать любую пару ячеек и поменять значения в этих ячейках.

Нужно за минимальное количество ходов получить матрицу, в которой будут выполнены оба условия:

- в каждой строке сначала идут нули, а потом единицы (строка может содержать только нули или только единицы);
- в каждом столбце сначала идут нули, а потом единицы (столбец может содержать только нули или только единицы).

Формат ввода

В первой строке даны два числа n и m ($1 \leq n, m \leq 100$) — размеры матрицы. В последующих n строках заданы по m символов, каждый из которых 0 или 1.

Формат вывода

Выведите целое число — минимальное количество обменов, для получения необходимой матрицы.

Пример 1

Ввод	Вывод
2 2 10 00	1

Пример 2

Ввод	Вывод
4 4 1000 0000 0000 0011	1

Пример 3

Ввод	Вывод
4 3 011 000 000 100	2

Примечания

В первом примере единицу из первой позиции первой строки следует поменять с нулем из второй позиции второй строки.

Во втором примере единицу из первой позиции первой строки можно поменять местами с последним нулем из третьей строки или вторым нулем из последней строки.

В третьем примере единицы из первой строки следует поменять местами с нулями из последней строки.

Язык GNU c++17 7.3

Набрать здесь

Отправить файл

1

Отправить

Предыдущая

Следующая

Algorithm 2021, Qualification

🕒 28 ноя 2021, 17:48:56
старт: 3 окт 2021, 14:12:29
финиш: 3 окт 2021, 16:12:29
длительность: 02:00:00
начало: 27 сен 2021, 12:00:00
конец: 3 окт 2021, 23:59:00

Ф. Переворот пути

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	2 секунды	1Gb	стандартный ввод или input.txt	стандартный вывод или output.txt
Python 3.7.3	7 секунд	1Gb		
Python 3.7 (PyPy 7.3.3)	7 секунд	1Gb		
C# (MS .Net 5.0)+ASP	3 секунды	1Gb		
Kotlin 1.4.30 (JRE 11)	3 секунды	1Gb		
OpenJDK Java 11	3 секунды	1Gb		
Golang 1.16	3 секунды	1Gb		

Решение, корректно работающее в ограничениях $n \leq 2000$, будет оценено в 2 балла.

Решение, корректно работающее для случая, когда максимальная степень вершины не превосходит 2, будет оценено в 2 балла.

Полное решение будет оценено в 7 баллов (включая 4 балла за подзадачи выше).

Дано дерево из n вершин. В нем выбрали две случайные различные вершины p и q (каждая пара вершин могла быть выбрана с одинаковой вероятностью) и перевернули путь между ними со всеми внутренними отрезками, не включая отрезки от самих p и q (смотрите примеры для лучшего понимания).

Найдите математическое ожидание расстояния между вершинами 1 и n .

Формат ввода


В первой строке дано целое число n ($2 \leq n \leq 200\,000$) — количество вершин в дереве.

В следующих $n - 1$ строках даны два числа u_i, v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$) — ребра дерева.

Формат вывода

Выведите целое число — математическое ожидание расстояния между вершинами 1 и n , умноженное на $\frac{n(n-1)}{2}$.


Пример 1

Ввод  Вывод 

Ввод  Вывод 


4
1 2
2 3
3 4
12

Пример 2

Ввод  Вывод 

4
1 2
2 3
2 4
10

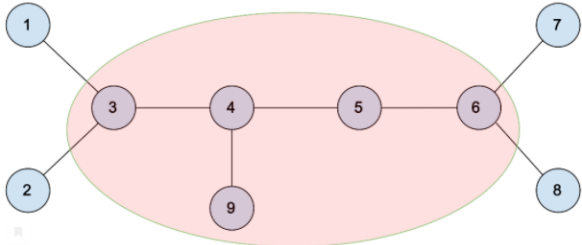
Пример 3

Ввод  Вывод 

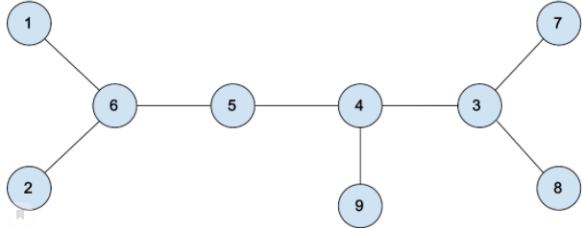
4
1 4
2 4
3 4
8

Примечания

Предположим, изначально дерево выглядело так:



Если $p = 3, q = 6$, то после переворота оно преобразуется в следующее дерево:



Обратите внимание, что ребро 4–9 осталось, так как это внутренний отрезок на пути между 3 и 6.