Программирование+ + настольные игры с ИКИТом

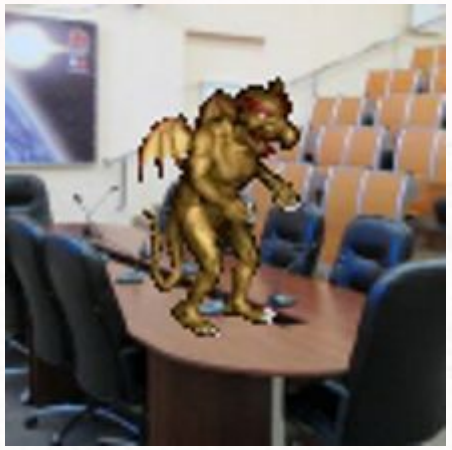
Выпуск №8

Разработчик: Халтурин Е.

Школа программиста

Минимальная стоимость проезда (332)

сложность



Ссылка

Минимальная стоимость проезда (332)

Будем определять минимальную стоимость проезда до [i] станции, где [i] меняется от 1 до n. Очевидно, что для [0] станции стоимость проезда до неё равна 0.

Для первой станции есть только один вариант доехать до неё, с нулевой станции.

Для второй станции уже два варианта, либо мы едем с нулевой станции, либо сначала оптимально до первой, а от неё уже на вторую.

Минимальная стоимость проезда (332)

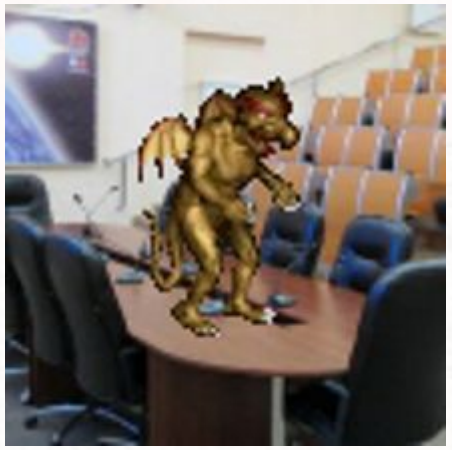
Пример:
4
7 10 20 38
4 8 10
2 12
10

Станции [i]: 0 1 2 3 4
Минимальная стоимость проезда: 0 7 10 12 **17**

CODEFORCES β
Sponsored by Telegram

Чунга-Чанга (1181A)

сложность



Ссылка

Чунга-Чанга (1181A)

Саша и Маша точно могут купить n = [x/z] + [y/z] кокосов. Если на оставшиеся деньги (то есть r = x mod z + y mod z) нельзя купить кокос (r<n), то ответ cout<<n<<“ “<<0;
Если можно, то нужно узнать, сколько нужно денег передать, то есть q = min(n- (x mod z), n-(y mod z)) и ответ cout<<n+1<<“ “<<q;

CODEFORCES β
Sponsored by Telegram

Разделение числа (1181B)

сложность

Ссылка

Разделение числа (1181B)

Найдем место разреза линии. Будем начинать делить строку от середины, и бежать указателем в разные стороны, пока символ строки равен ‘0’ (так как число при разделении не имеет ведущих нулей).
Найдем первое корректное разделение в левую и первое корректное разделение в правую сторону. Может быть и такая ситуация, что корректных разделений в одну из сторон нет, это тоже нужно учесть (пример: 1000010 не имеет корректных разделений в левую сторону).

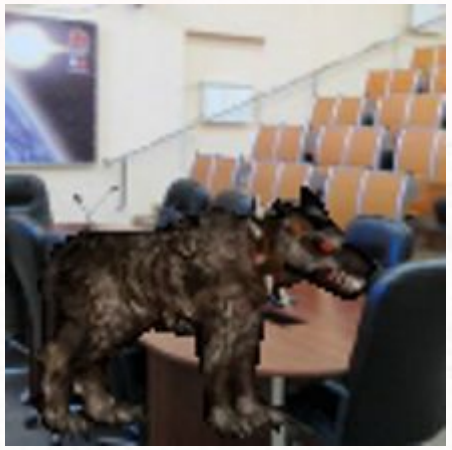
Разделение числа (1181B)

После этого, делим строку на две (можно воспользоваться функцией substr) и складываем их как числа (на python это удобнее, так как там реализована длинная арифметика).
Проверяем, какое число меньше, при делении в левую или при делении в правую сторону. Меньшее и будет

CODEFORCES β
Sponsored by Telegram

Флаг (1181C)

сложность

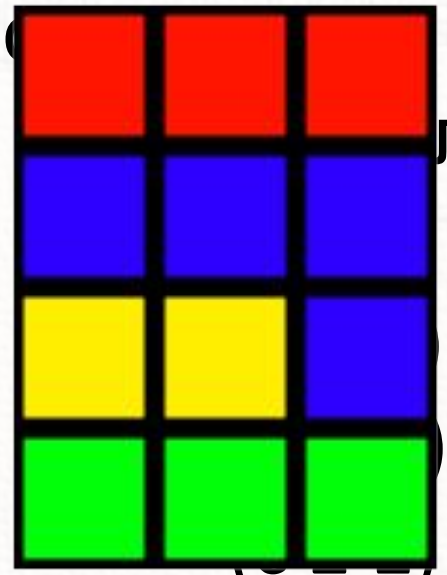


Ссылка

Флаг (1181C)

Динамическое программирование:

Будем хранить в каждой позиции матрицы помимо символа также информацию о начальной позиции, когда полоска в высоту одного и того же цвета заканчивается, а также, сколько флагов можно составить, если они имеют правый нижний угол в данной по

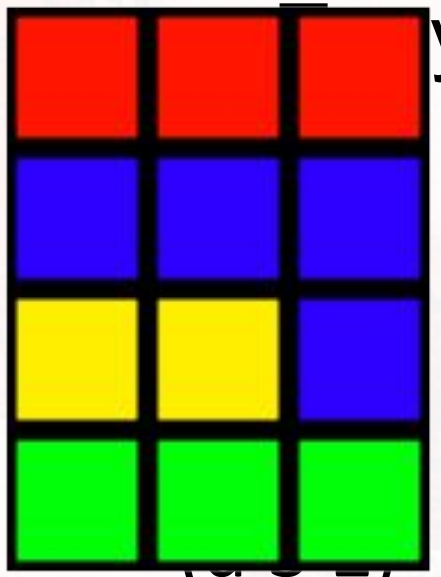


Пример: лученная матрица:

4 3

aaa	(a 0 **0**)	(a 0 **0**)
bbb	(b 1 **0**)	(b 1 **0**)
ccb	(c 2 **2**)	(b 1 **0**)

Флаг (1181C)

Пример:  ученная матрица:

4 3

aaa	(a 0 **0**)	(a 0 **0**)
bbb	(b 1 **0**)	(b 1 **0**)
ccb	(c 2 **2**)	(b 1 **0**)
ddd	(d 3 **2**)	(d 3 **0**)

Итоговый ответ будет сумма всех ответов в каждой ячейки.

Как быстро находить частичные ответы? Нужно проверить, можно ли получить флаг, рассматривая только текущий столбец. Если можно, то смотрим, чтобы слева от текущей позиции были бы такие же цвета, и с такой же высотой полоски. Если это так,

Флаг (1181C)

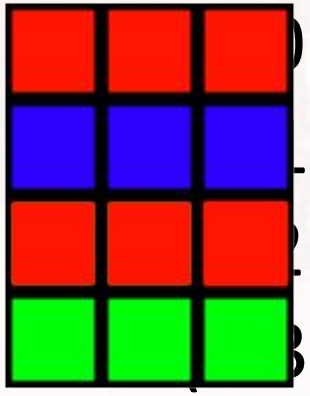
4 3 Полученная матрица: Ответ:12

aaa (a 0 **0**) (a 0 **0**) (a 0 **0**)

bbb (b 1 **0**) (b 1 **0**) (b 1 **0**)

aaa (a 2 **1**) (a 2 **2**) (a 2 **3**)

ddd (d 3 **1**) (d 3 **2**) (d 3 **3**)



7 3 Полученная матрица: Ответ:7

cdq (c 0 0) (d 0 0) (q 0 0)

cec (e 0 0) (e 1 0) (c 1 0)

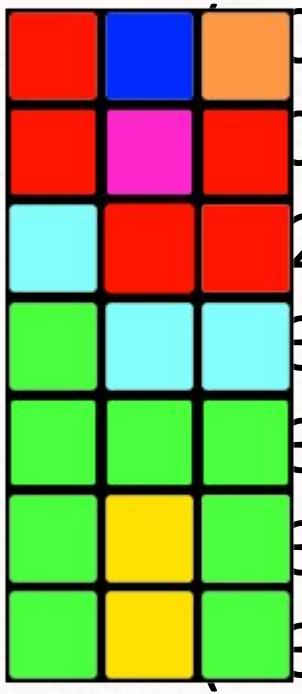
bcc (b 2 0) (c 2 1) (c 1 0)

abb (b 3 1) (b 3 1) (b 3 0)

aaa (a 3 0) (a 4 1) (a 4 2)

afa (f 3 0) (f 5 1) (a 4 0)

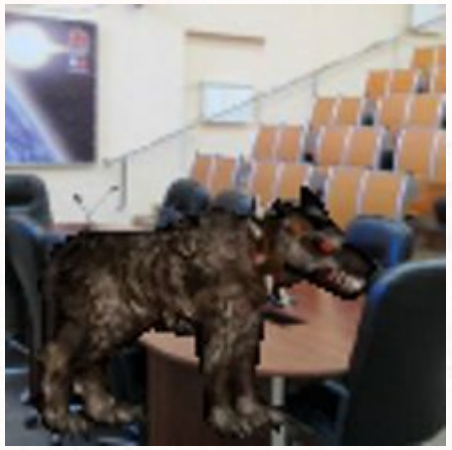
afa (f 3 0) (f 5 0) (a 4 0)



CODEFORCES β
Sponsored by Telegram

Нажатия на кнопки (102168G)

сложность



Ссылка

Нажатия на кнопки (102168G)

Для случая, когда одна или две кнопки, решим задачу отдельно. Будем рассматривать случай, когда $n \geq 3$.
Пусть s1 – сумма всех чисел до нажатий, а s2 – после нажатий.
Введем обозначения. Для [i] кнопки: a[i] – число на кнопке до нажатий, b[i] – число на кнопке после нажатий, x[i] – количество нажатий на [i] кнопку.

Нажатия на кнопки (102168G)

Рассмотрим систему уравнений для n = 3.

$$\begin{cases} a[1] - x[1] + x[2] + x[3] = b[1] \\ a[2] + x[1] - x[2] + x[3] = b[2] \\ a[3] + x[1] + x[2] - x[3] = b[3] \end{cases}$$

Решим систему: Пусть $s = x[1] + x[2] + x[3]$, тогда

$$\begin{cases} a[1] + s - b[1] = 2 * x[1] \\ a[2] + s - b[2] = 2 * x[2], \text{ и при этом, если сложить все равенства:} \\ a[2] + s - b[2] = 2 * x[2] \end{cases}$$

$$s1 + s * (n - 2) = s2 \Rightarrow s = \frac{s2 - s1}{(n-2)}$$

Программирование+ + настольные игры с ИКИТом

Текст задач взят с сайтов:

- acmp.ru
- codeforces.com
- informatics.mccme.ru

Выпуск №8

Разработчик: Халтурин Е.