Московская командная олимпиада 2010, лига А

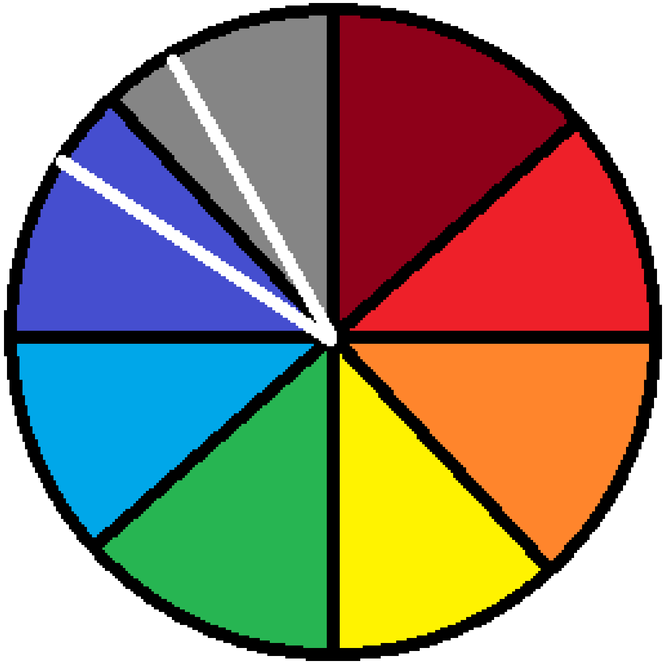
Разбор задач

A. Цветы
Автор: научный комитет
Разработчик: Оля Вечкасова

- Максимальное количество цветов в букете равно k = C div A (целочисленное деление)
- Какие из k ромашек можно заменить на гладиолусы?
- p = (C – k*A) div (B – A), если p <= k
- Ответ:

min(k, (C – k*A) div (B – A))*(B – A) + k*A

B. Пицца
Автор: Владимир Гуровиц
Разработчик: Василевский Борис



- M = K, ответ M: резать с небольшим сдвигом
- M < K, ответ M: нельзя попадать по границам начинки. Если попали – можно немного сдвинуть
- M > K, ответ K: симметричная ситуация
- Исключение K = 1, ответ 0

C. Решение задач
Автор: Михаил Пядёркин
Разработчик пожелал остаться неизвестным

- Если задачу можно решить – сделаем это, и хуже не станет.
- Жадный алгоритм: пока можем что-нибудь решить, решаем. Работает за O(N^2).
- Решать можно в порядке возрастания a[i].
- Сортировка задач по a[i] и линейный проход: O(N log N).

D. Стильная одежда
Автор и разработчик:
Глеб Евстропов

- Существует две различные идеи решения.
- Первая. Отсортируем по цвету элементы одежды каждого типа (по отдельности), заведём по указателю для списков каждого типа.
- Очередной шаг: из четырёх значений выберем позицию с минимальным цветом, увеличим соответствующий указатель.
- На каждом шаге рассматриваем разницу между максимальным и минимальным цветом.

- Вторая (авторское решение). Изменим формулировку: на прямой находятся точки разного типа, найти отрезок минимальной длины, содержащий все 4 типа (координатой точки является цвет).
- f(i, j) = число различных типов на отрезке с i по j.
- g(j) = максимальное i (1 <= i <= j), такое что f(i, j) = 4. Если такого i не существует, g(j) = 0.
- f(i, j + 1) >= f(i, j), следовательно g(j + 1) >= g(j).
- Заведем два указателя и будем последовательно, сдвигая правую границу, сдвигать левую пока это возможно.

E. Подтасовка результатов

Автор идеи и разработчик: Михаил Пядёркин

Автор задачи: Елена Андреева

- Обозначим за r – количество участников, которые ниже Васи в обоих турах, за l – количество участников выше Васи в обоих турах.
- Если наше место не в пределах от l+1 до n-r, то ответа не существует. Иначе будем его строить. Занизим всем баллы:

Баллы	Кто – 1 тур	Кто – 2 тур
400		
399		
…		
N	A_1	B_1
N-1	A_2	B_2
…	…	…
1	A_N	B_N

- Пусть при такой таблице мы занимаем место p, а нужно занять большее место. Докажем, что мы можем так поправить эту таблицу, чтобы занять любое место от p-1 до максимально возможного.

- Будем по одному поднимать баллы другим:

Баллы	Кто – 1 тур	Кто – 2 тур
400	A_1	B_1
399	…	B_2
…		…
N		
N-1	A_2	
…	…	…
1	A_N	B_N

- Каждый раз еще один человек становится строго выше нас, т.к. 400-max(n-i, n-j)+1 (мин. балл при нахождении в верхней части) > 400-(N-1)+1 > 202 > N-1+1+1 > i+1+j+1 (наш балл).
- При каждом подъеме место меняется не более чем на один. Всего мы можем изменить место на минимально возможное (всех поднять над нами) => можем получить любое между.

- Аналогично, рассмотрим таблицу, где баллы расставлены начиная с максимальных. В ней мы занимаем то же самое место р. Начнем опускать вниз людей. Каждый раз наше место меняется не более чем на один, в конце у нас максимально возможное место => можем получить любое между.
- Значит для каждого тура можно считать, что ответ – две части, одна прижата к верху, другая к низу.

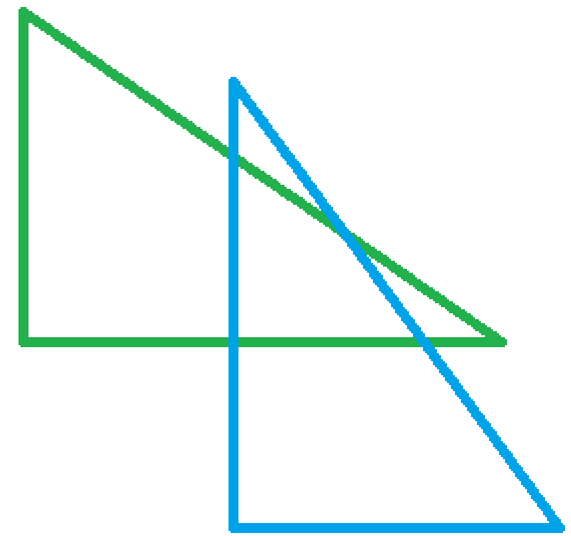
Решение

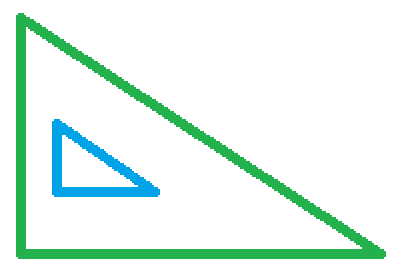
- Переберем деление в первом туре
- Переберем деление во втором туре.
- Расставим баллы: в каждом туре верхняя половина деления прижата к верхнему краю, нижняя – к нижнему.
- Проверим, получилось ли нужное место.
- Возможны реализации за различную сложность

F. Прямоугольные треугольники
Автор задачи и разработчик:
Михаил Пядёркин

Решение для общего случая

- Задача: проверить, пересекаются ли два произвольных треугольника
- Проверить на пересечение стороны различных треугольников.
- Остается случай вложенных треугольников. Достаточно проверить, что одна из вершин одного из треугольников лежит внутри другого.

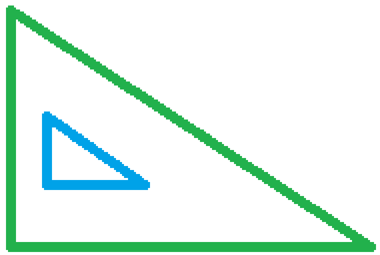




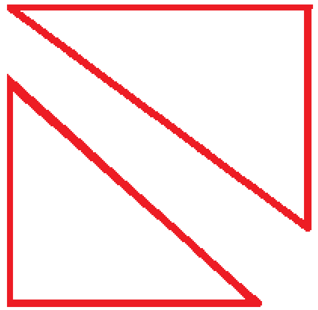
Решение 2а

- Заметим, что если прямоугольные треугольники не вложены и пересекаются, то у них обязательно пересекаются катеты.
- Задача свелась к проверке пересечения вертикальных/горизонтальных отрезков и проверке принадлежности точки прямоугольному треугольнику

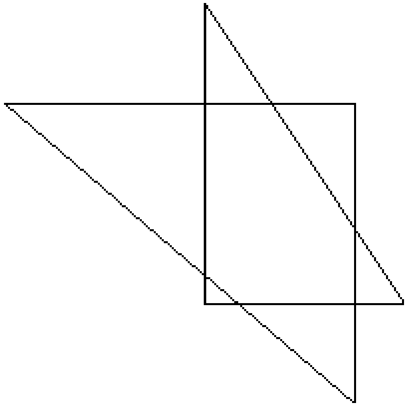
Test #9



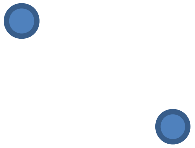
Test #10



Test #15



Test #65



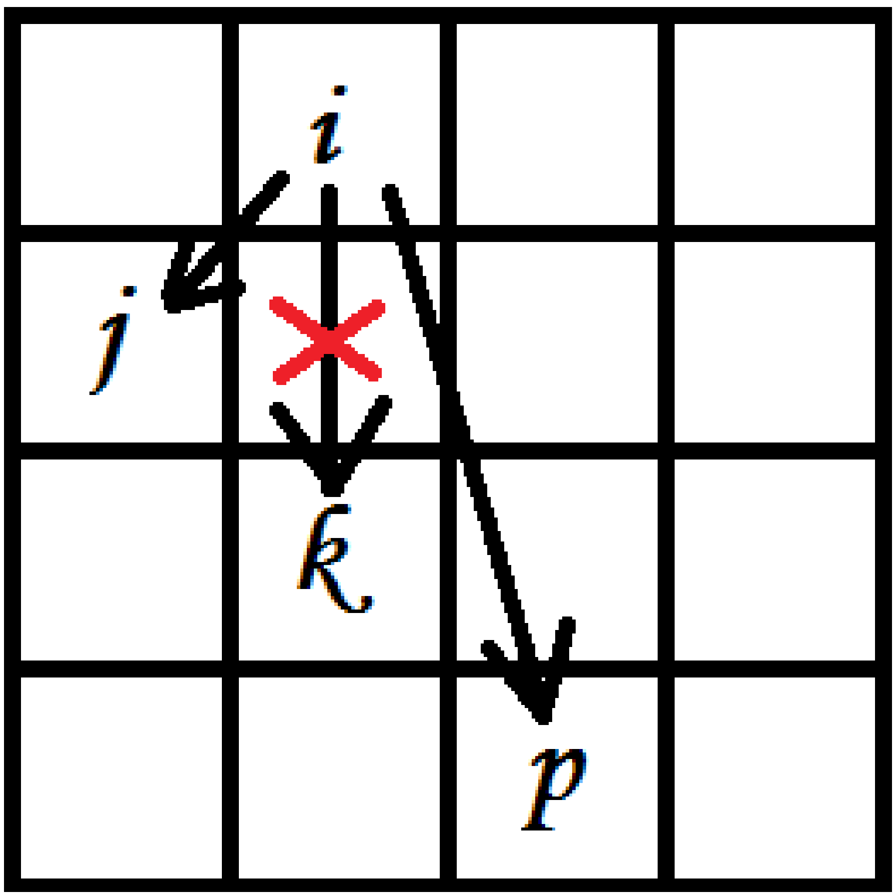
G. Словарь
Автор: Владимир Гуровиц
Разработчик: Кирилл Афанасьев

- V[i][j] – j-е латинское слово в переводе i-го английского; W – искомый словарь.
- L – отсортированный массив латинских слов.
- Пробежимся по V, для каждого V[i][j] нужно:
  – Найти k: L[k] = V[i][j] бинарным поиском
  – Добавить в список W[k] английское слово i.
- W автоматически отсортирован.
- Аккуратный ввод и вывод – самая сложная часть. Удобно читать всю строку целиком.

Н. Пешки

Авторы: Фёдор Ивлев и Глеб Евстропов

разработчик: Глеб Евстропов



- Граф. Множество вершин — начальные позиции пешек и фигур, ребро (i, j) существует, если пешка находясь в позиции i может пройти вперед и побить позицию j, не обращая внимание на то что стоит на пути.

- Рассмотрим покрытие графа минимальным числом вершинно-непересекающихся путей. Их не менее P.
- Если их ровно P, то существование ответа, доказывается индукцией по строкам.
- Если их больше чем P, то ответа не существует. Пусть он существует, тогда по нему однозначно восстанавливается покрытие мощности P.

I. Трамвай
Автор: Михаил Густокашин
Разработчик: Сергей Рогуленко

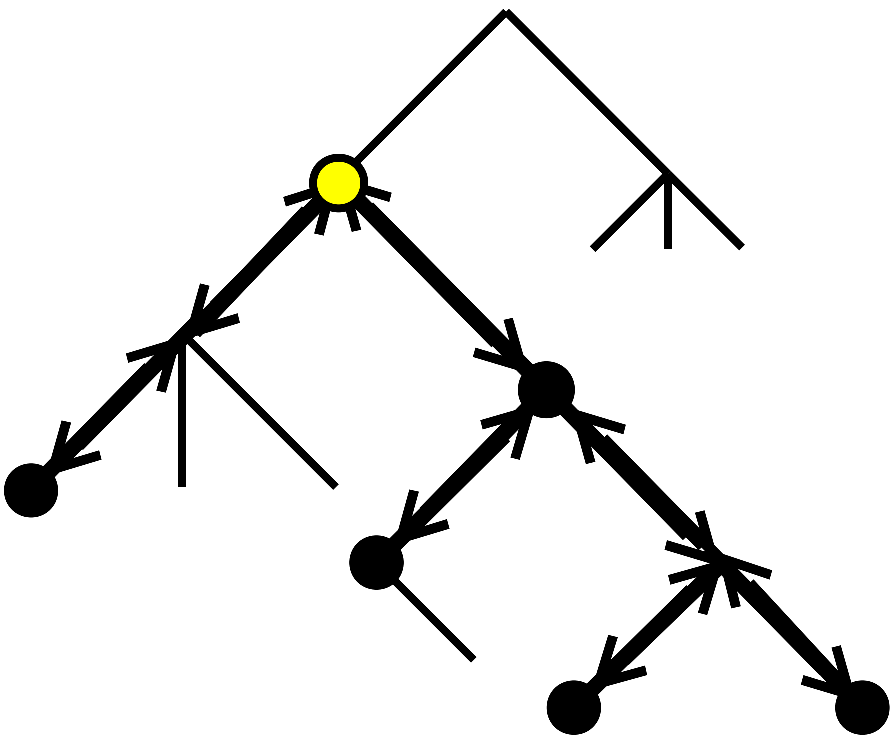
- Пусть A[i][j] – минимальное время на преодоление первые i остановок, пройдя j метров пешком.
- По A[i][*] вычисляем A[i + 1][*]:
- Промежуток номер i можно либо проехать, либо пройти. Пусть

  D = a[i + 1] – a[i],   next = min(K, j + D)

- Идём: A[i][j] + D/v → A[i + 1][next].
- Едем: A[i][j] + D/w + wait(A[i][j] – (a[i] – a[1])/w) → A[i + 1][j],
- где wait(S) – сколько придётся ждать трамвая на первой остановке, если с начала прошло S минут.

J. Путешествия в реальности
Автор: Петр Михеев
Разработчик: Антон Полднев

- **Ответ:** удвоенный суммарный вес всех рёбер, лежащих на пути от стартовой вершины до любой помеченной.



- Вместо стартовой вершины можно рассмотреть наименьший общий предок помеченных вершин (старая стартовая считается помеченной).

Реализация 1. Подъём вверх

- Из каждой помеченной вершины поднимаемся вверх, помечая рёбра, пока не попадём либо в корень, либо в вершину, принадлежащую одному из уже помеченных рёбер.
- Найти наименьший общий предок всех помеченных вершин и не учитывать рёбра выше него.
- По каждому ребру проходим ровно один раз $\Rightarrow$ сложность — $O(n)$.

Реализация 2. ДП на дереве

- Пусть $f(v)$ — удвоенная сумма весов рёбер, лежащих на пути от вершины v до какой-нибудь помеченной вершины из поддерева v.
- $f(v) = \sum_{P_i = v} f(i) + 2 \cdot \sum_{P_j = v} \left(T_j - T_v\right)$ для всех i и тех j, в поддереве которых есть помеченные вершины.
- Ответ — наименьшее $f(v)$ для тех v, в поддереве которых есть все помеченные.
- Каждое ребро рассматриваем один раз $\Rightarrow$ сложность — $O(n)$.