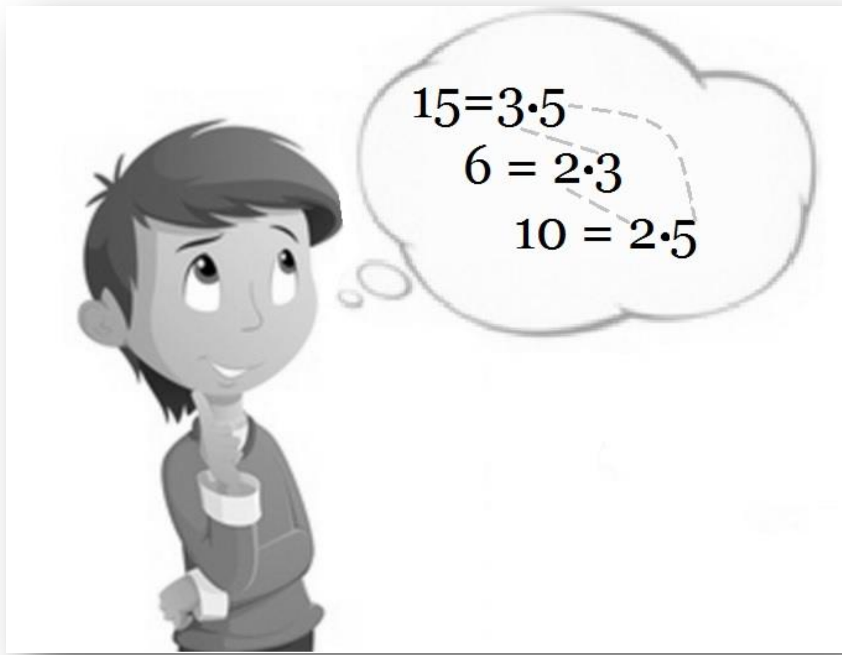
Разбор заключительного этапа республиканской олимпиады по информатике 2016г.

Тур 1 Задача 1
Занимательная нумерология



Сложность: низкая

Условие:

- Дано количество доступных чисел ($2 \le N \le 10^5$) и число ($2 \le K \le N$)
- Необходимо выбрать K различных чисел от 1 до N таких, что у любых двух из них НОД больше 1.
- Если это невозможно, нужно вывести - 1.

Частичные ограничения:

- $K = 1$, $N \le 20$ – не менее 20 баллов
- $K \le 5$, $N \le 20$ – не менее 40 баллов
- $N \le 20$ – не менее 60 баллов
- $N \le 5000$ – не менее 80 баллов

Решение:

- Если K > N/2, то ответ -1
  - Подсказка: НОД(X, X+1) = 1
- Иначе достаточно вывести числа 2, 4, 6, …, 2K

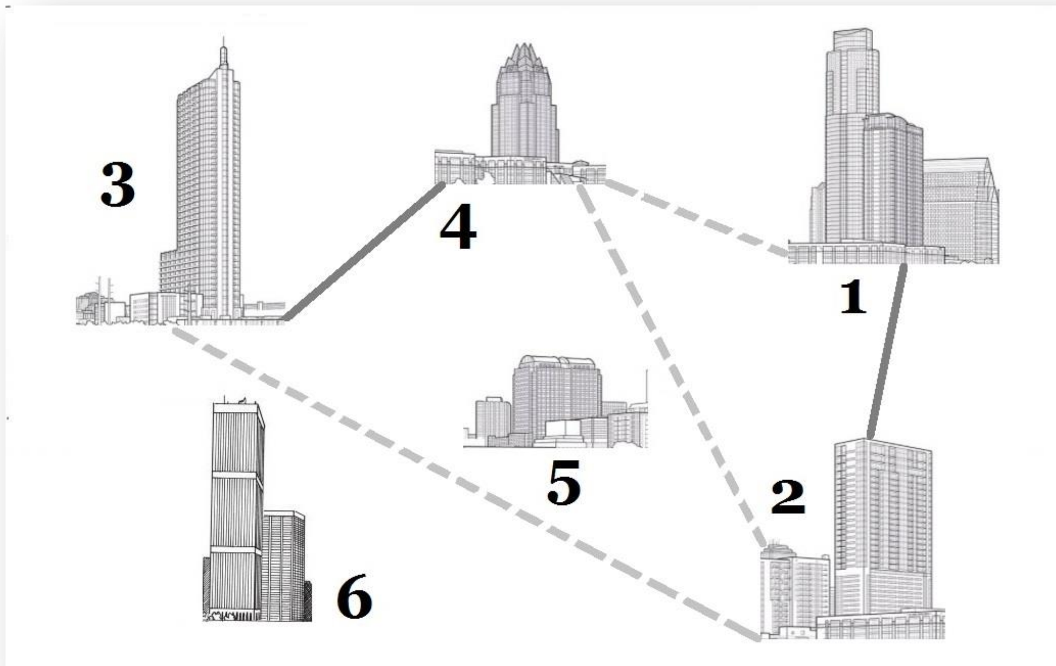
Challenge:

- Предложите алгоритм, который за приемлемое время проверяет последовательность на соответствие условию

Результаты:


Распределение баллов
Количество участников
120
100
80
60
40
20
0
0
0
0
0
1
1
0
1
3
0
114
0-9
10-19
20-29
30-39
40-49
50-59
60-69
70-79
80-89
90-99
100
баллы

Тур 1 Задача 2
Дороги Байтландии



Сложность: средняя-

Условие:

- Задан граф на N ($2 \le N \le 10^5$) вершинах с M ($0 \le M \le 10^5$) рёбрами

- Нужно добавить ещё K ($1 \le K \le \min(10^9, (N-1)N/2-M)$ новых рёбер так, чтобы граф содержал минимальное или максимальное возможное количество компонент связности

Частичные ограничения:

- $N \le 10$, $M \le 10$, $K \le 1$ – не менее 40 баллов
- $N \le 50$, $M \le 100$, $K \le 2$ – не менее 60 баллов
- $N \le 1000$, $M \le 10000$ – не менее 80 баллов

Решение:

- Находим компоненты связности графа и их размеры (поиск в ширину или в глубину, или система не пересекающихся множеств)

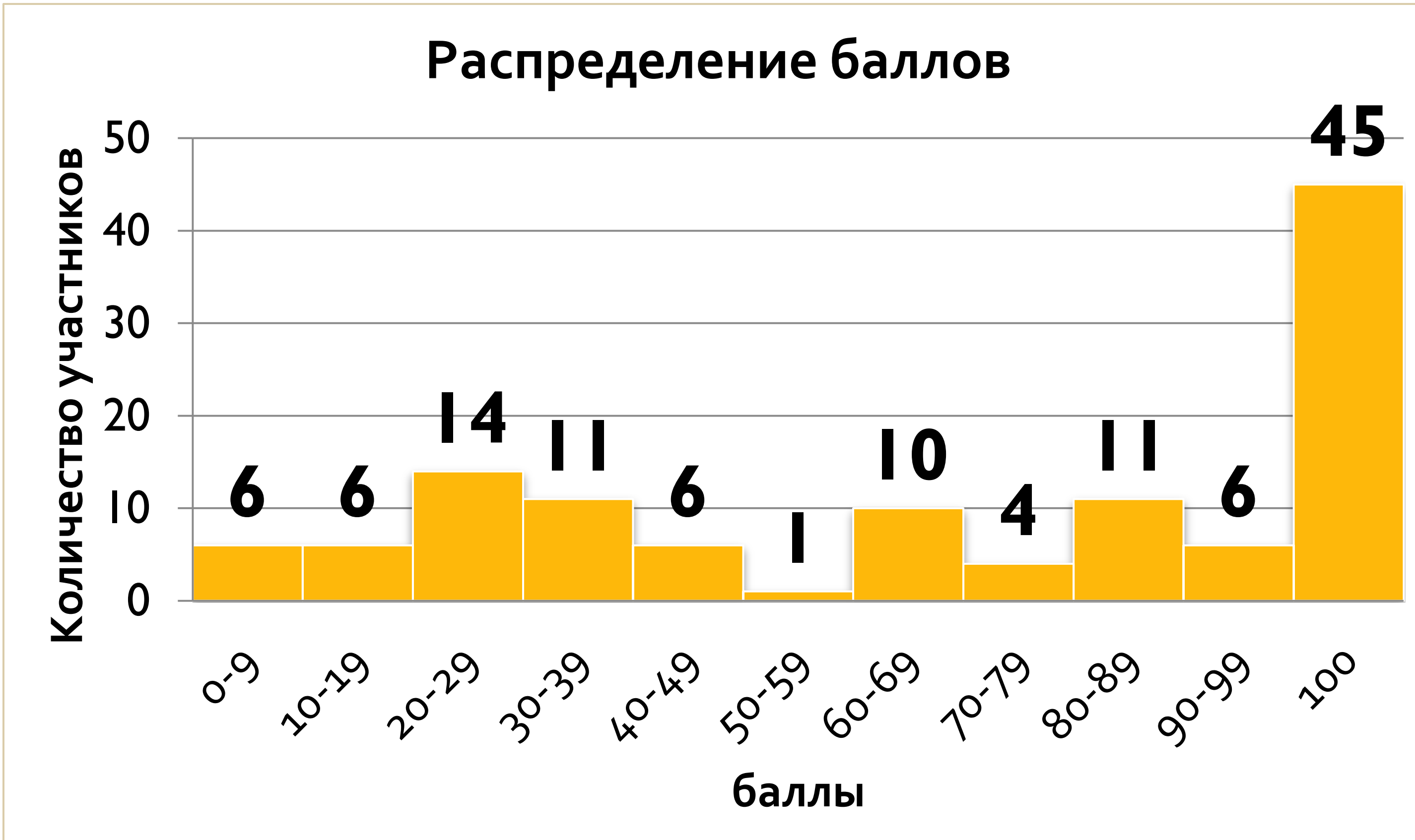
- Чтобы получить минимальное возможное количество компонент связности соединяем любую вершину первой компоненты с любой вершиной второй, любую вершину второй с любой вершиной третьей и т.д. Тогда, если K меньше, чем количество компонент связности (H), то ответ H-K, иначе ответ 1.

Решение:

- Чтобы получить максимальное возможное количество компонент связности нужно как можно больше рёбер добавить внутри компонент. Если в компоненте T вершин и D рёбер, то можно добавить $T(T-1)/2-D$ новых рёбер

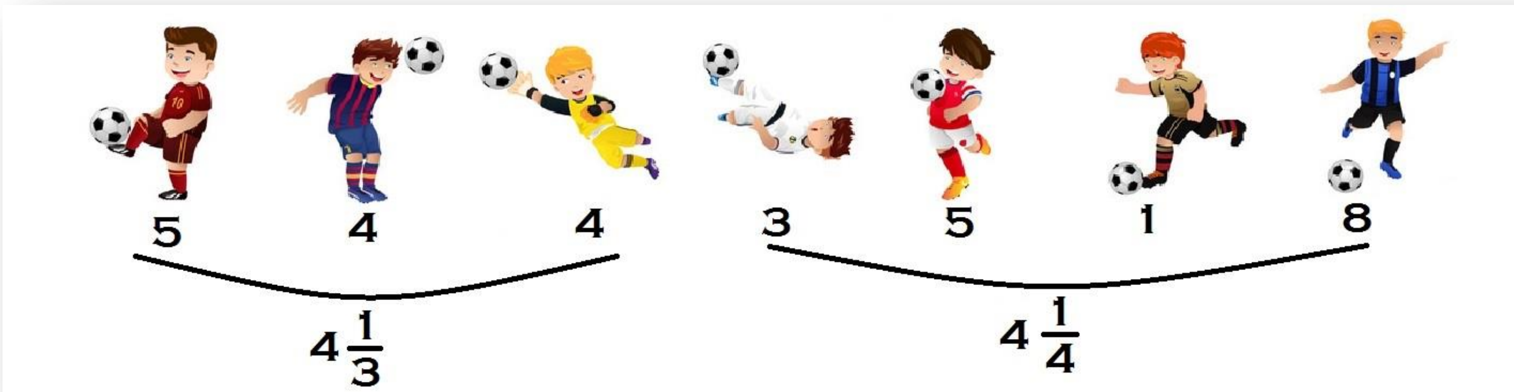
- Если добавили меньше K рёбер, то нужно добавлять рёбра между компонентами (от каждой вершины одной компоненты к каждой вершине другой). Это следует делать предварительно отсортировав компоненты связности в порядке убывания их размеров

Результаты:



Распределение баллов
Количество участников
0
10
20
30
40
50
6
6
14
11
6
1
10
4
11
6
45
0-9
10-19
20-29
30-39
40-49
50-59
60-69
70-79
80-89
90-99
100
баллы

Тур 1 Задача 3
Команды



Сложность: высокая

Условие:

- Дана последовательность из N ($6 \le N \le 10^4$) целых чисел A_i ($1 \le A_i \le 10^9$)
- Необходимо разбить эту последовательность на K ($2 \le K \le 500$) непрерывных частей размера не меньше, чем M ($3 \le M$)
- Для каждой из частей последовательности вычисляется среднее арифметическое, и минимальная из этих величин должна быть как можно больше

Частичные ограничения:

- $N \le 50$, $A_i \le 10$, $K = 2$ – не менее 20 баллов
- $N \le 50$, $A_i \le 1000$, $K \le 4$ – не менее 40 баллов
- $N \le 100$, $A_i \le 1000$ – не менее 60 баллов
- $N \le 3000$ – не менее 80 баллов

Решение:

- Научимся для фиксированного значения X определять можно ли разбить последовательность таким образом, чтобы среднее арифметическое всех групп было не меньше чем X.

- Заметим, что следующие две системы равносильны:

Решение:

$$\begin{cases} \dfrac{A_1 + A_2 + \cdots + A_{T_1}}{T_1} \geq X \\ \dfrac{A_{T_1+1} + A_{T_1+2} + \cdots + A_{T_1+T_2}}{T_2} \geq X \\ \cdots \\ \dfrac{A_{\sum_{i=1}^{K-1} T_i+1} + A_{\sum_{i=1}^{K-1} T_i+2} + \cdots + A_{\sum_{i=1}^{K} T_i}}{T_K} \geq X \end{cases}$$

$$\Leftrightarrow$$

$$\begin{cases} (A_1 - X) + (A_2 - X) + \cdots + (A_{T_1} - X) \geq 0 \\ (A_{T_1+1} - X) + (A_{T_1+2} - X) + \cdots + (A_{T_1+T_2} - X) \geq 0 \\ \cdots \\ \left(A_{\sum_{i=1}^{K-1} T_i+1} - X\right) + \left(A_{\sum_{i=1}^{K-1} T_i+2} - X\right) + \cdots + \left(A_{\sum_{i=1}^{K} T_i} - X\right) \geq 0 \end{cases}$$

Решение:

- Задача свелась к проверке того, можно ли разбить последовательность $B_i = A_i - X$ на K последовательных групп размера не менее M так, чтобы в каждой группе сумма была неотрицательной

Проверка за O(NK):

- Можно вычислить:

$$F_{ij} = \begin{cases} 1, \text{если можно разбить } i \text{ первых элементов на } j \text{ групп} \\ 0, \text{если нельзяразбить } i \text{ первых элементов на } j \text{ групп} \end{cases}$$

- Для этого вычислим префиксные суммы для B, $S_i=B_1+B_2+\ldots+B_i$

- **Тогда F_{ij} = 1 тогда и только тогда, когда существует такое $k \le i-M$, что $F_{k,j-1}=1$ и $B_{k+1}+\ldots+B_i \ge 0$ ($S_i-S_k \ge 0 \Leftrightarrow S_i \ge S_k$).** Положим $F_{0,0}=1$.

Проверка за O(NK):

- **Тогда F_{ij} = 1 тогда и только тогда, когда существует такое $k \le i-M$, что $F_{k,j-1}=1$ и $B_{k+1}+\ldots+B_i \geq 0$ ($S_i-S_k \geq 0 \Leftrightarrow S_i \geq S_k$)**

- Можно данное утверждение переписать в виде $\mathrm{Min}(S_t) \le S_i$, $t \le i - M$, $F_{t,j-1}=1$

Нам нужно с «задержкой» в M шагов обновлять минимальное значение S_t для таких t, что на «предыдущем слое» $j-1$ $F_{t,j-1}=1$

Если $F_{N,K}=1$, то для заданного X можно разбить последовательность A на группы, удовлетворяющие условию задачи

Проверка за O(NlogN):

- Будем вычислять F_i, равное максимальному количеству групп, на которое можно разбить первые i элементов последовательности B_i

- **Отметим, что если мы можем разбить на j групп, то можем разбить и на любое меньшее количество, объединив «лишние» группы в одну**

Проверка за O(NlogN):

- Для подсчёта F_i снова вычислим префиксные суммы S_i.
- Для удобства будем считать $S_0=0$ и $F_0=0$.
- Далее нужно на всех префиксах $0 \le j \le i-M$ и с префиксными суммами, не превосходящими S_i, найти максимум F_j.
- Тогда $F_i=\max(F_j)+1$ или $Fi=-\infty$, если все $S_j>S_i$.

Проверка за O(NlogN):

Далее будем действовать следующим способом:

1) Отсортируем массив пар (S_i, i) лексикографически
2) Будем обрабатывать индексы i в порядке сортировки пар. Это позволит избавиться от условия $S_j \le S_i$. Т.е. необходимо на отрезке от 0 до i-M найти максимальное (уже вычисленное) значение F_j. Для этого можно использовать такие структуры данных как «Дерево Фенвика» или «Дерево отрезков».
3) Если $F_N \ge K$, то для заданного X можно разбить последовательность A на группы, удовлетворяющие условию задачи

Завершение

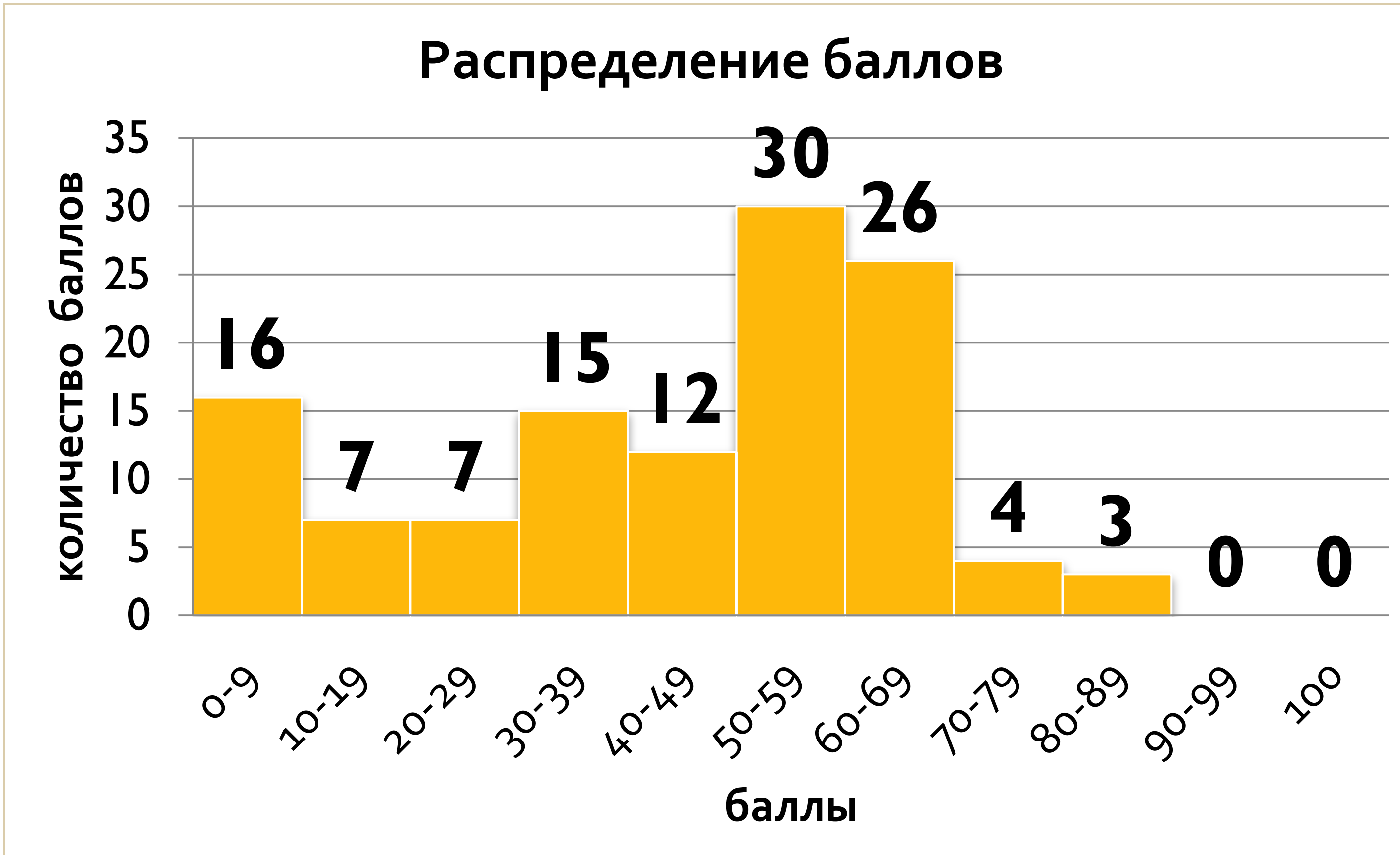
Осталось определить оптимальное значение X.

Для этого воспользуемся двоичным поиском на интервале $[0, \max(A_i)]$.

Чтобы показать, что двоичный поиск будет работать корректно, заметим, что обе наши проверки говорят «хорошо», если существует хотя бы одно разбиение на группы, удовлетворяющее ограничению на X. Очевидно, что до оптимального значения $X = X_0$ подходит хотя бы «оптимальное разбиение» на группы.

Заметим, что средние значения в группах могут отличаться на достаточно малую величину. Например, при специально подобранных значениях A_i, разность между средними значениями может быть равна $1/(G^2-G)$, где $G \le N/2$. Это необходимо учитывать при реализации двоичного поиска.

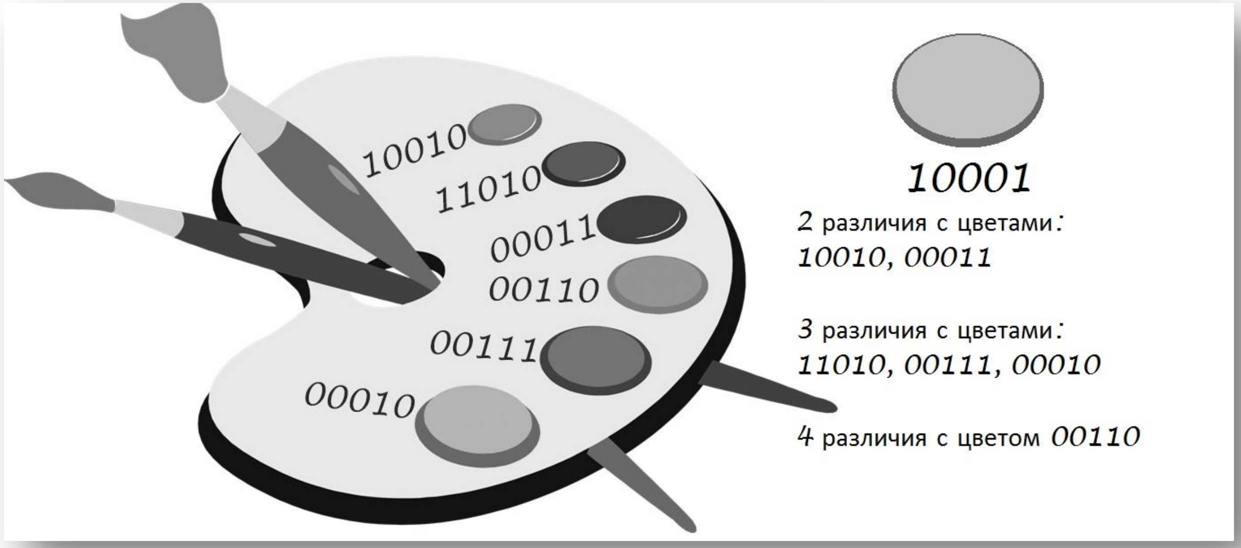
Результаты:


Распределение баллов
количество баллов
35
30
25
20
15
10
5
0
16
7
7
15
12
30
26
4
3
0
0
0-9
10-19
20-29
30-39
40-49
50-59
60-69
70-79
80-89
90-99
100
баллы

Challenge:

- Эту задачу можно решить в целых (рациональных числах). Кто-то может что-то предложить?

Тур 1 Задача 4
Палитра



Сложность: высокая

Условие:

- Изначально есть пустое множество
- С течением времени в него добавляются числа, не превосходящие 2^{20}
- Нужно отвечать на запросы «какое число ближайшее к заданному?», где под степенью близости понимается количество совпадающих битов в их двоичной записи

Частичные ограничения:

- $K \le 10$, $N \le 1000$ – не менее 40 баллов
- $K \le 10$, $N \le 100\,000$ – не менее 60 баллов
- $K \le 15$, $N \le 300\,000$ – не менее 70 баллов
- $K \le 18$ – не менее 80 баллов

Решение:

- Заведём массив D, где D_i есть наибольшая степень близости числа i с каким-либо числом, находящимся во множестве, а пусть массив C хранит сами эти числа

- При добавлении числа i запускаем обход в ширину из числа i, изначально положив $D_i = 0$

- На итерации обхода в ширину пытаемся изменить 1 бит в текущем числе X, получаем числа $Y_1, Y_2, \ldots, Y_K$. Также обновляем D для этих чисел значением $D_X + 1$, если оно строго меньше, чем текущее

Оценка сложности:

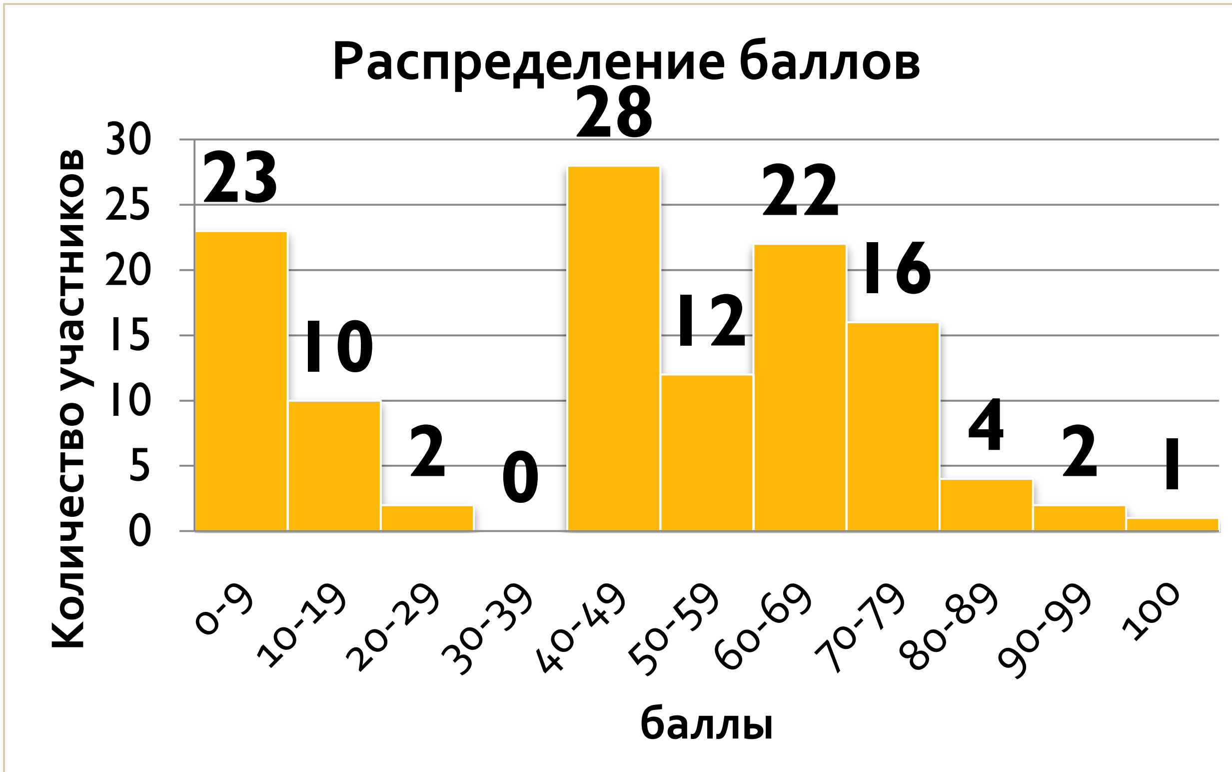
- Заметим, что $0 <= D_i <= K$.

- Также значение D_i при обновлении может только уменьшаться.

- Дадим оценку количеству обновлений: $2^K K^2$

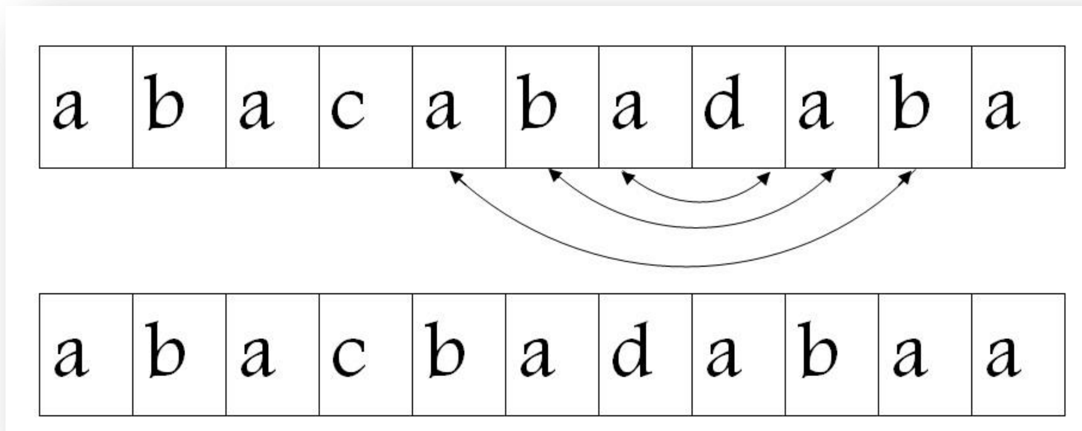
- Таким образом указанное решение имеет асимптотику $O(2^K K^2)$ и укладывается в ограничения по времени

Результаты:



Распределение баллов
Количество участников
30
25
20
15
10
5
0
23
10
2
0
28
12
22
16
4
2
1
0-9
10-19
20-29
30-39
40-49
50-59
60-69
70-79
80-89
90-99
100
баллы

Тур 2 Задача 1
Защита авторских прав



Сложность: низкая

Условие:

- Даны две строки, состоящие из строчных латинских букв. Обе строки имеют длину N ($1 \le N \le 10^6$)
- Нужно проверить, можно ли получить первую строку из второй переворотом какой-либо подстроки

Частичные ограничения:

- N ≤ 10 – не менее 20 баллов
- N ≤ 255 – не менее 40 баллов
- N ≤ 5000 – не менее 60 баллов
- N ≤ 100 000 – не менее 80 баллов

Решение:

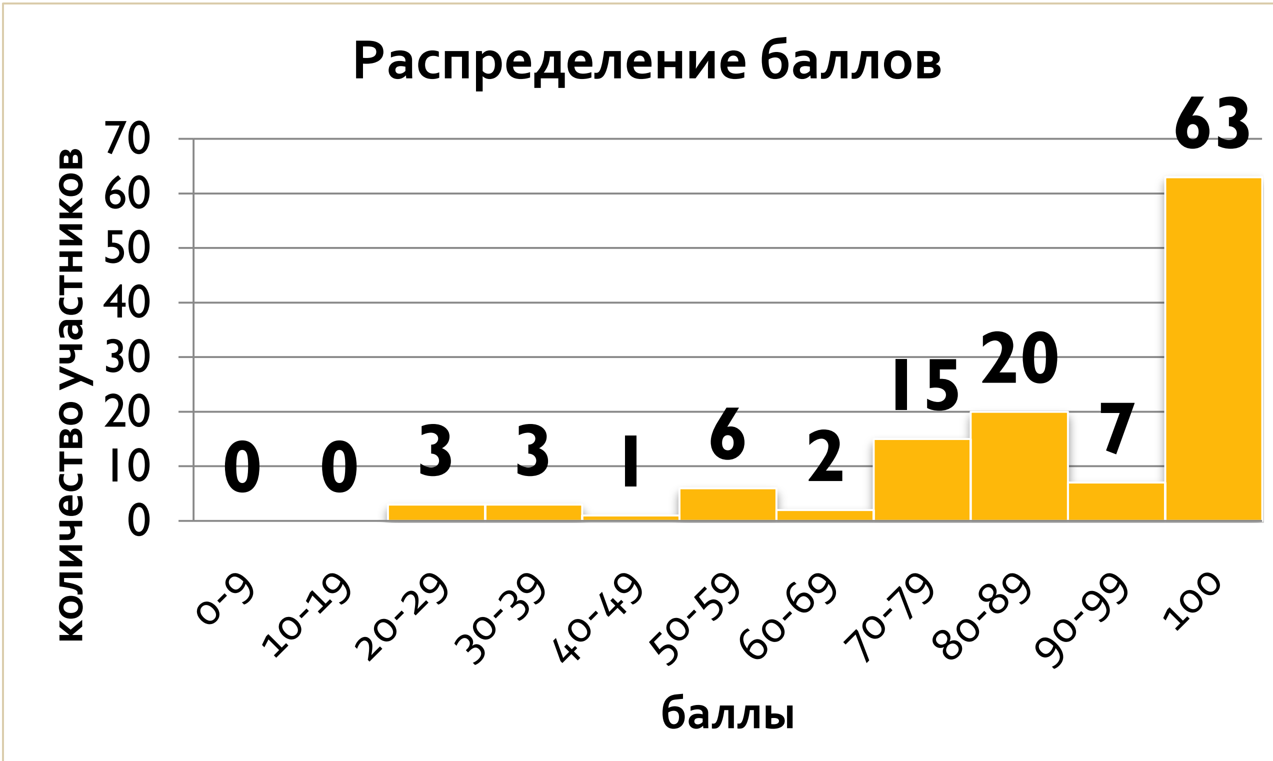
- Если строки равны, сертификатом является любая пара (i, i), соответствующая подстроке длины 1

- Найдем длину совпадающего префикса P и совпадающего суффикса S

- Пробуем перевернуть подстроку P+1 .. N-S

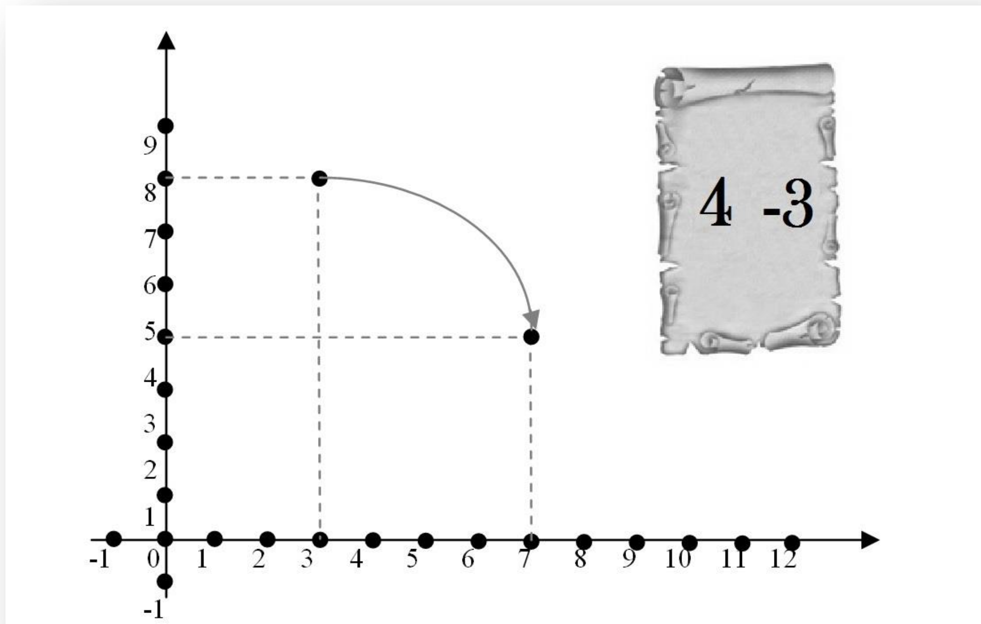
- Если после этого, строки совпали, то сертификатом является (P+1, N-S), иначе решения нет

Результаты:



Распределение баллов
количество участников
70
60
50
40
30
20
10
0
0
0
3
3
1
6
2
15
20
7
63
0-9
10-19
20-29
30-39
40-49
50-59
60-69
70-79
80-89
90-99
100
баллы

Тур 2 Задача 2
Магические свитки



Сложность: средняя

Условие:

- Задана точка на плоскости, с координатами по модулю не превосходящими 10^9. Также задано N ($N \le 10^5$) векторов (X_i, Y_i) ($-10^9 \le X_i, Y_i \le 10^9$)
- Требуется выбрать произвольное количество векторов так, чтобы, отложив их сумму от точки (0, 0), мы получили точку максимально удалённую от заданной
- В задаче рассматривается Манхэттенское расстояние

Частичные ограничения:

- $N \le 20$, $X_i = 0$ или $Y_i = 0$ – не менее 20 баллов
- $N \le 1000$, $X_i = 0$ или $Y_i = 0$ – не менее 40 баллов
- $N \le 5000$ – не менее 60 баллов

Решение:

- Запишем Манхэттенское расстояние:

$$|X-C| + |Y-D|$$

- Раскроем модули. Это можно сделать четырьмя способами

$$X-C > 0,\ Y-D > 0$$
$$X-C \leq 0,\ Y-D > 0$$
$$X-C > 0,\ Y-D \leq 0$$
$$X-C \leq 0,\ Y-D \leq 0$$

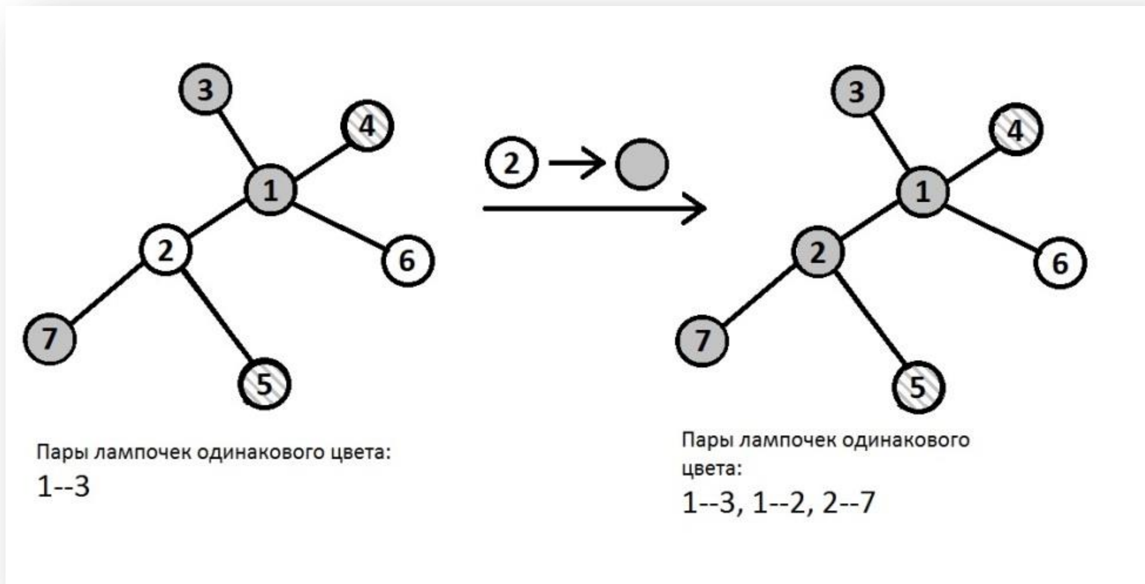
Решение:

- После того, как мы выбрали знаки, мы можем записать сумму, которую хотим оптимизировать
- Стоит выбирать вектора, которые увеличивают нашу сумму
- По итогу выбираем лучший из четырёх ответов

Результаты:

Распределение баллов
количество участников
35
30
25
20
15
10
5
0
14
12
13
3
31
5
6
1
5
2
28
0-9
10-19
20-29
30-39
40-49
50-59
60-69
70-79
80-89
90-99
100
баллы

Тур 2 Задача 3 Праздничное украшения



Сложность: высокая

Условие:

- Дано дерево на N ($1 \le N \le 300\,000$) вершинах, у которого каждая вершина покрашены в один из K($1 \le K \le 10^9$) цветов

- M ($1 \le M \le 300\,000$) раз производится перекрашивание вершины

- После каждого перекрашивания нужно сообщить количество рёбер, обе концевые вершины которых имеют один цвет

Частичные ограничения:

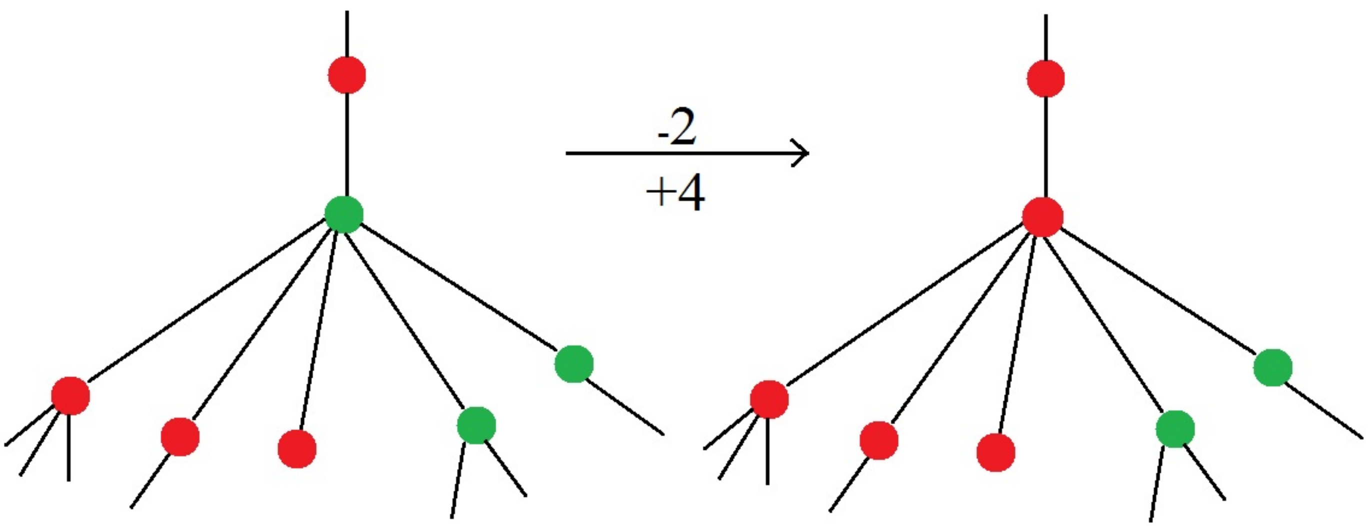
- $N \le 100$, $M \le 100$, $K \le 100$ – не менее 20 баллов
- $N \le 5000$, $M \le 100$, $K \le 100$ – не менее 40 баллов
- $N \le 10^5$, $M \le 10^5$, $K \le 100$ – не менее 60 баллов
- $N \le 10^5$, $M \le 10^5$ – не менее 80 баллов

Решение:

- В самом начале посчитаем количество одноцветных рёбер

- Теперь, каждый раз, когда перекрашивается вершина, нужно отнять количество одноцветных рёбер, в которое входит эта вершина, перекрасить её (будем хранить цвета в массиве), и добавить новое количество одноцветных рёбер, в которые входит эта вершина

Решение:



Решение:

- Задача свелась к тому, чтобы для вершины быстро узнать, сколько у неё есть соседей заданного цвета

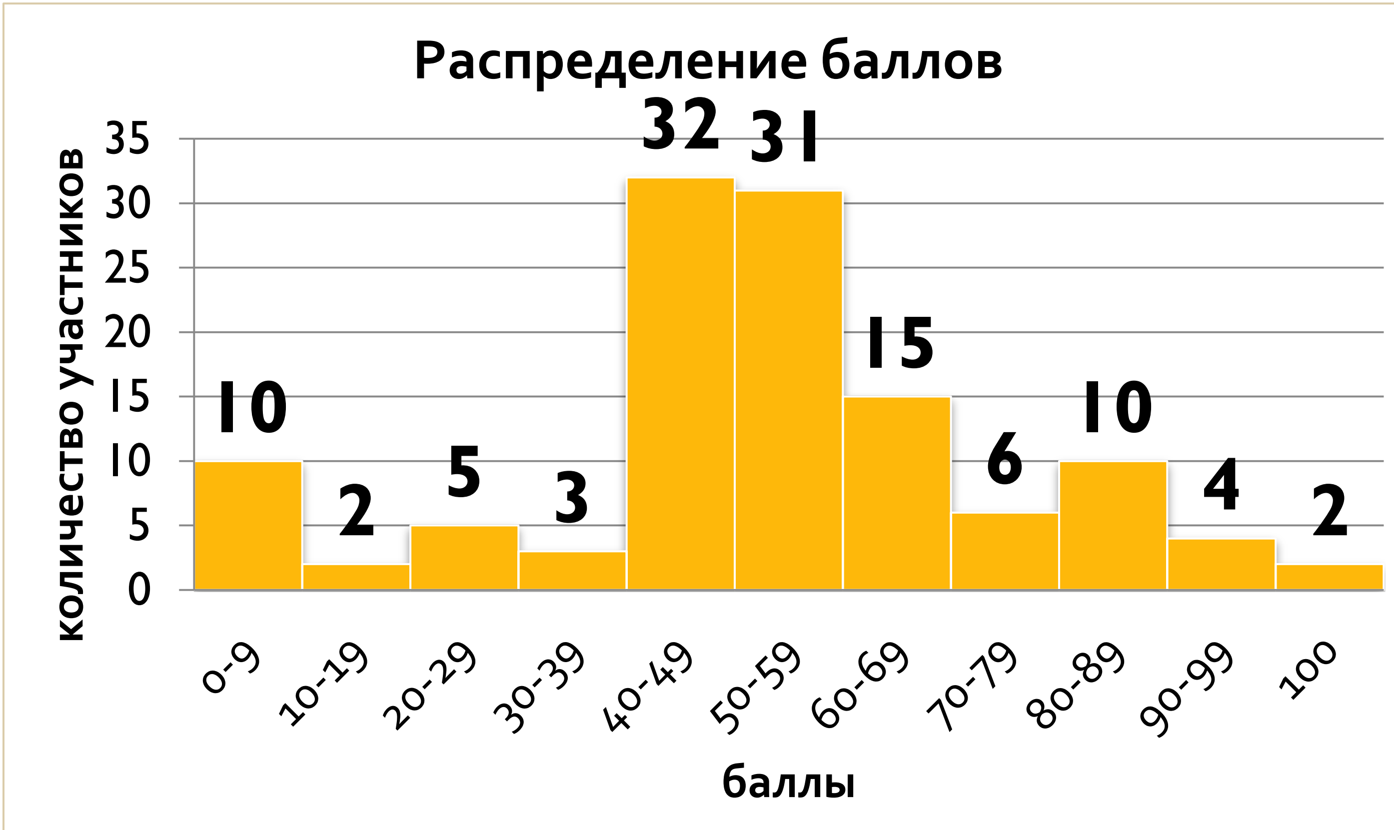
- Если мы изначально подвесим дерево, то можно отдельно рассматривать предка, отдельно потомков

- Цвет предка можем узнать просто посмотрев в массив цветов

Решение:

- Количество потомков определённого цвета можно поддерживать с помощью какой-либо структуры данных (массивы, словари)
- Тогда мы быстро сможем узнать старое и новое количество одноцветных рёбер с нашей вершиной
- При перекраске нашей вершины нужно не забыть обновить структуру данных её предка

Результаты:


Распределение баллов
количество участников
35
30
25
20
15
10
5
0
10
2
5
3
32
31
15
6
10
4
2
0-9
10-19
20-29
30-39
40-49
50-59
60-69
70-79
80-89
90-99
100
баллы

Challenge:

- Предложите алгоритм, который за приемлемое время решает аналогичную задачу для произвольного графа

Тур 2 Задача 4 Общежитие

#	#	#	#	#	#	#	#	#	#	#
#	.	#	B	#	9	#	8	#	.	#
#	.	.	.	.	.	.	.	.	.	#
#	.	#	6	#	1	#	7	#	.	#
#	.	#	#	#	#	#	#	#	.	#
#	.	#	5	#	0	#	6	#	.	#
#	.	.	.	.	.	.	.	.	.	#
#	.	#	3	#	2	#	4	#	.	#
#	#	#	#	#	#	#	#	#	#	#

Сложность: высокая

Условие:

- Дан план общежития, в котором клетки ‘#’ являются запрещёнными, а по всем остальным клеткам можно перемещаться

- В некоторых клетках есть числа

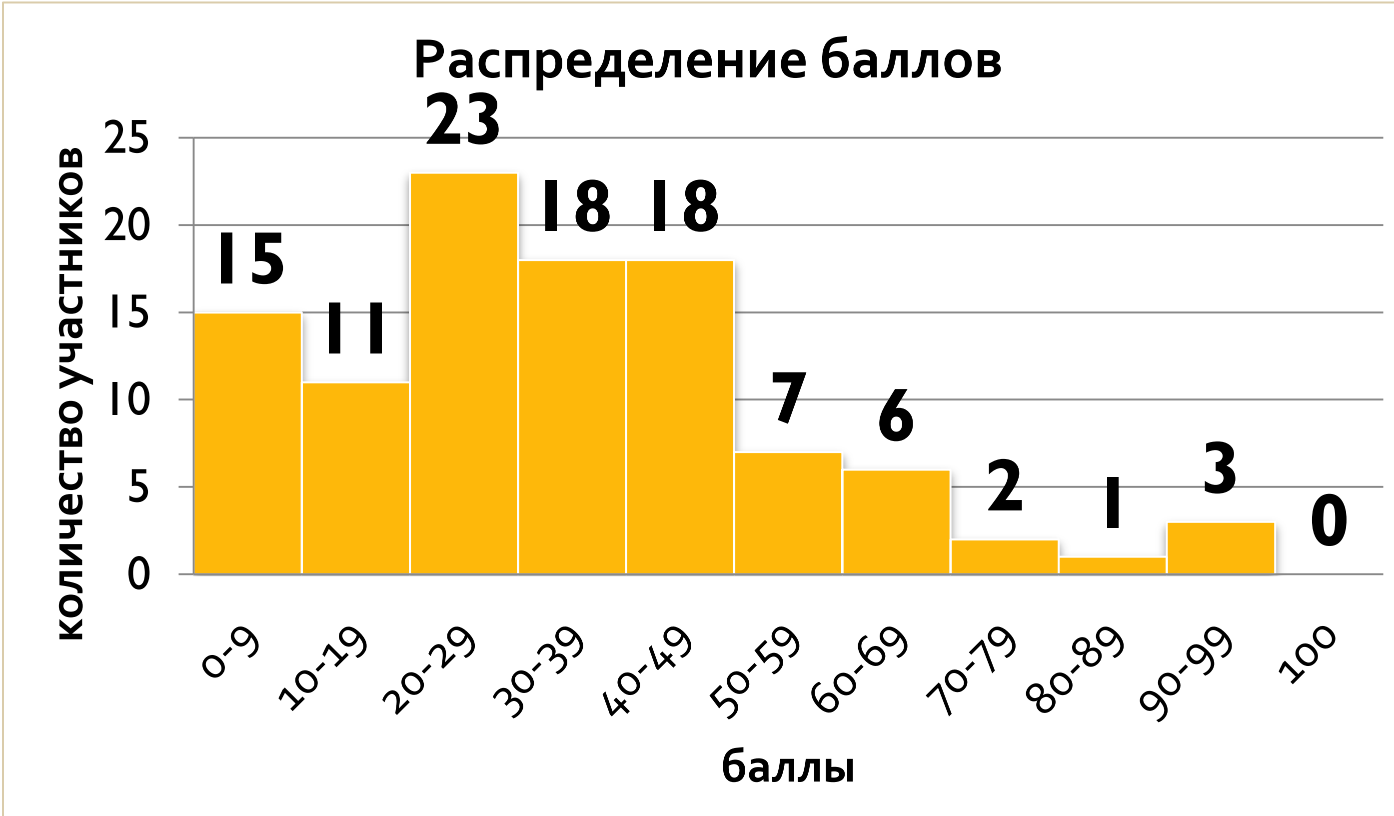
- Надо за указанное время выйти из заданной клетки, пройтись по разрешённым клеткам, собрать как можно большую сумму чисел и вернуться назад

- В одну и ту же клетку с числом нельзя заходить дважды

Решение:

- RandomWalk
- RandomWalk с запоминанием соседа
- Жадный RandomWalk + GridSearch
- Другие эвристики

Результаты:



Распределение баллов
количество участников
25
20
15
10
5
0
15
11
23
18
18
7
6
2
1
3
0
0-9
10-19
20-29
30-39
40-49
50-59
60-69
70-79
80-89
90-99
100
баллы

Результаты олимпиады:

СПАСИБО ЗА ВНИМАНИЕ