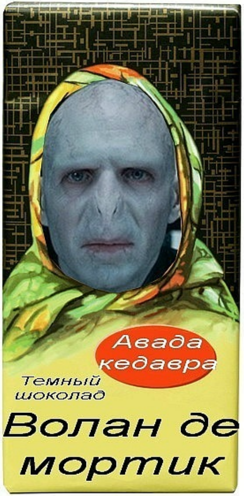
ИОИП 2013:
Разбор задач

17 марта 2013 года

Задача «Гарри Поттер и нос Волан-де-Морта»



Над задачей работали

- Идея задачи: Андрей Станкевич
- Текст условия: Павел Кротков
- Тесты, проверяющая программа и др.: Павел Кротков
- Текст разбора: Павел Кротков

Формулировка задачи

- Дано клетчатое поле размера $n \times m$
- Некоторые клетки заняты
- Сколькими способами на этом поле можно разместить прямоугольник размера 1×2, не задев занятых клеток?

Идея решения

- Для каждой клетки проверим, может ли она быть верхней клеткой в вертикально размещенном прямоугольнике, или левой клеткой в горизонтально размещенном прямоугольнике
- Заметим, что мы проверили все возможные позиции прямоугольника

Код решения

```
for i := 1 to n do
    for j := 1 to m do begin
        if (a[i][j] = '#') then continue;
        if ((i < n) and (a[i + 1][j] = '.')) then
            ans := ans + 1;
        if ((j < m) and (a[i][j + 1] = '.')) then
            ans := ans + 1;
    end;
```

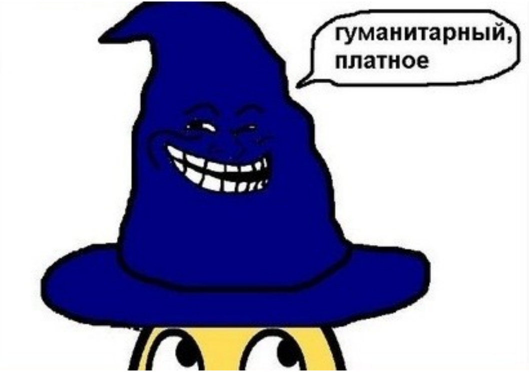
Частичные решения

- 40 баллов набирает решение, работающее для поля без занятых клеток
- в случае пустого поля ответ равен $2 \times (n-1) \times (m-1) + (n-1) + (m-1)$

Вопросы

- Вопросы?

Задача «Гарри Поттер и Распределяющая Шляпа»



Над задачей работали

- Идея задачи: Демид Кучеренко
- Текст условия: Андрей Комаров
- Тесты, проверяющая программа и др.: Андрей Комаров
- Текст разбора: Андрей Комаров

Формулировка задачи

- Задача с самой запутанной формулировкой
- В задаче требовалось вывести n-е число в бесконечной последовательности, создаваемой шляпой, по модулю p
- Начало последовательности: $0, 1, 1, 2, 1, 2, 2, 3, 1, 2, 2, 3, 2, 3, 3, 4, 1, 2, 2, 3, \ldots$

Формулировка задачи

- Последовательность создаётся следующим образом:
- Сначала есть только одно число — ноль: (0)
- На каждой итерации существующая последовательность выписывается дважды, после чего к каждому числу во второй половине прибавляется 1: $(0) \to (0, 1) \to (0, 1, 1, 2) \to (0, 1, 1, 2, 1, 2, 2, 3)$

Идея решения

- Найдём число, стоящее на n-м месте
- Пусть оно стало доступно на i-м шаге, то есть $2^{i-1} < n \leq 2^i$
- Оно получилось из числа на $(n - 2^{i-1})$-м месте прибавлением единицы
- Найдём ответ для $(n - 2^{i-1})$ таким же способом

Код решения

```
pow2[0] := 1;
for i := 1 to 62 do
    pow2[i] := 2 * pow2[i - 1];
ans := 0;
for i := 62 downto 0 do
    if n > pow2[i] then begin
        n := n - pow2[i];
        inc(ans);
    end;
```

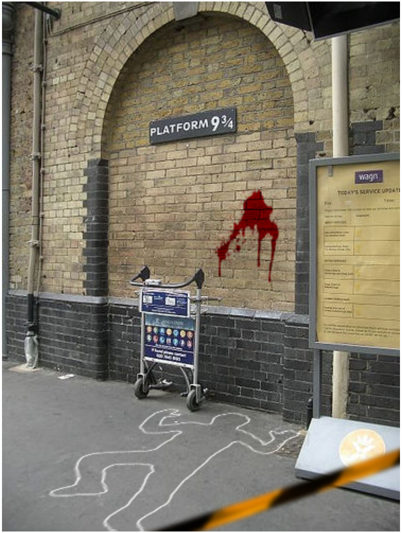
Частичные решения

- 30 баллов можно было набрать, если просто сделать то, что написано в условии — для каждого запроса считать все последовательности и брать n-е число из первой, имеющей длину хотя бы n
- 60 баллов можно было набрать, если заметить, что ответ не сильно зависит от p — можно было один раз создать последовательность нужной длины и выдавать в качестве ответа на очередной запрос `a[n] mod p`

Вопросы

- Вопросы?

Задача «Гарри Поттер и железная дорога»



Над задачей работали

- Идея задачи: Борис Минаев
- Текст условия: Виталий Демьянюк
- Тесты, проверяющая программа и др.: Виталий Демьянюк
- Текст разбора: Виталий Демьянюк

Формулировка задачи

- Дан связный граф, содержащий n вершин и m ребер. Надо занумеровать ребра так, чтобы для каждой вершины наибольший общий делитель (НОД) номеров ребер, соединяющих эту вершину с остальными, был равен единице.
- Если решения нет, то нужно вывести «`IMPOSSIBLE`».

Идея решения

- Если в графе больше одной вершины со степенью один, то решения не существует.
- Будем решать задачу с помощью поиска в глубину. В качестве начальной вершины выберем ту, у которой степень равна единице (если такой нет, выберем любую).
- Будем нумеровать ребра в порядке их первого просмотра в поиске в глубину.

Объяснение решения

- У первой просмотренной вершины будет ребро с номером один, поэтому НОД всех ребер, выходящих из нее, будет равен единице.
- Степень всех остальных вершин больше единицы. Разница между номером ребра, по которому мы впервые пришли в вершину, и номером первого просмотренного из этой вершины ребра равна единице. Значит, для этой вершины НОД всех инцидентных ей ребер тоже будет равен единице.
- Так как граф связный, мы пройдем по всем вершинам, и поэтому НОД ребер в каждой вершине будет равен единице.

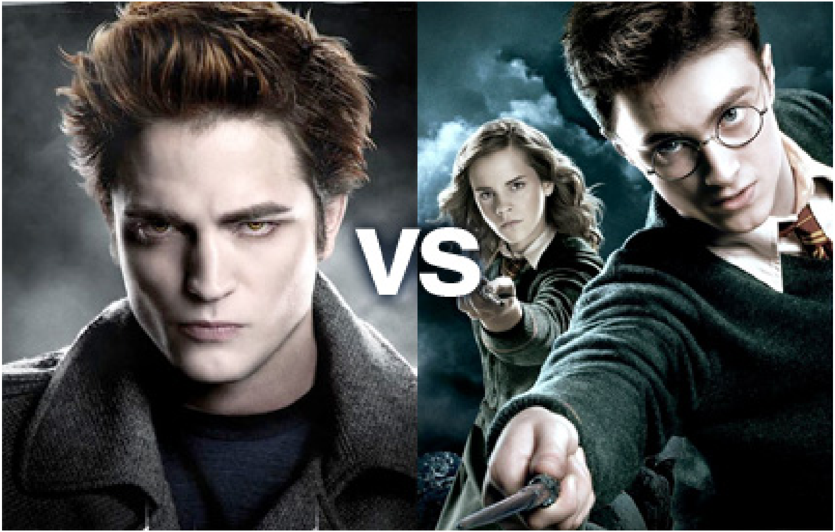
Частичные решения

- Полный перебор набирает 20 баллов.
- Поиск в глубину, реализованный на матрице смежности вместо списков смежности, набирает 60 баллов.

Заключение

- Время работы $\mathcal{O}(n + m)$.
- Вопросы?

Задача «Гарри Поттер и Зал Пророчеств»



Над задачей работали

- Идея задачи: Николай Ведерников
- Текст условия: Николай Ведерников
- Тесты, проверяющая программа и др.: Николай Ведерников
- Текст разбора: Николай Ведерников

Формулировка задачи

- Дан массив из $2n$ чисел
- Требовалось разбить на 2 массива по n чисел следующим образом:
  - Выбираются два наиболее близких числа, если таких несколько, то те два числа, у которых сумма наибольшая
  - Затем самый большой элемент из пары записываем в первый массив, а второе число — во второй

Идея решения - 1

- Отсортируем массив за $\mathcal{O}(n \times \log(n))$
- Заметим, что два наиболее близких числа, являются соседними в отсортированном массиве
- Построим двусвязный список по отсортированному массиву, а так же создадим приоритетную очередь (например, с помощью `TreeSet` в `Jave`, или с помощью `set` в `C++`) на тройке чисел разница между соседними, суммой чисел и указателем на первый элемент в списке

Идея решения - 2

- Находим минимум в приоритетной очереде, по найденому элементу получаем два наиболее близких числа и указатели на эти числа в списке
- Удаляем эти числа из списка и соотвествующие им элементы в приоритетной очереди
- При удалении из списка, возможно появилась новая пара соседних чисел, тогда добавляем в приоритетную очередь новый элемент

Время работы

- Сортировка массива за $\mathcal{O}(n \times \log(n))$
- Удаление одной пары чисел из нашей структуры происходи за $\mathcal{O}(\log(n))$, таких операций у нас $\mathcal{O}(n)$
- Итоговое время работы $\mathcal{O}(n \times \log(n))$

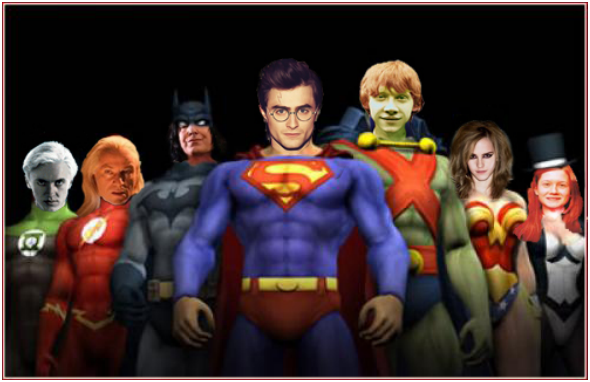
Частичные решения

- 20 баллов набирает решение, которое каждый раз ищет в массиве два самых близких элемента
- 40 баллов набирает решение, которое предварительно сортирует массив и удаляет два соседних элемента за $\mathcal{O}(n)$

Вопросы

- Вопросы?

Задача «Гарри Поттер и битва за Хогвартс»



Над задачей работали

- Идея задачи: Артем Васильев
- Текст условия: Артем Васильев
- Тесты, проверяющая программа и др.: Артем Васильев
- Текст разбора: Павел Кротков

Формулировка задачи

- Дана перестановка чисел от 1 до n
- Сделано k итераций сортировки «пузырьком»
- Как после этого выглядит перестановка?

Идея решения

- Получим по перестановке «таблицу инверсий»
- Посмотрим, как она изменилась
- Получим по измененной таблице инверсий измененную перестановку

Таблица инверсий

- Таблица инверсий перестановки — последовательность чисел $a_1, a_2, \ldots, a_n$
- a_i — количество чисел таких, что они больше числа i, но стоят в перестановке левее числа i

Как вычислить?

- Идем по перестановке слева направо
- Каждый раз необходимо посчитать количество уже пройденных чисел, больших, чем текущее число в перестановке
- Необходимо считать сумму на отрезке
- Можно использовать, например, дерево Фенвика

Изменение таблицы инверсий

- Изучим сначала изменение за один проход сортировки
- Заметим, что ни один элемент не может увеличиться
- Если слева от числа i в перестановке были числа большие, чем оно само, то ровно одно из них (максимальное) после прохода окажется правее числа i
- Значит, все ненулевые элементы «таблицы инверсий» уменьшатся ровно на один

Изменение за несколько проходов

- Все элементы таблицы меньшие, чем k, обнулились
- Все остальные элементы таблицы уменьшились ровно на k

Восстановление ответа

- Будем расставлять числа в перестановку по убыванию, начиная с n
- Место очередного числа i — $(a_i + 1)$-я свободная позиция в текущей версии перестановки
- Проще всего использовать бинарный поиск и дерево Фенвика

Время работы

- Время работы описанного алгоритма составляет $\mathcal{O}(n \times \log^2(n))$
- Восстановление ответа можно делать несколько по-другому, использовав дерево отрезков
- В этом случае время работы составит $\mathcal{O}(n \times \log(n))$

Альтернативное решение

- Заметим, что ни один элемент не уйдет влево больше, чем на k позиций
- Положим первые k элементов перестановки в упорядоченное множество
- После этого будем добавлять в множество очередной элемент, а в ответ дописывать минимальный элемент из множества
- Когда добавлять будет нечего — перестанем добавлять, будем только дописывать в ответ

Частичные решения

- Решение, делающее то, что написано в условии, получало 40 баллов

Спасибо за внимание!

- Спасибо за внимание!
- Вопросы?