

Алгоритмы и структуры данных

Задание к лабораторной работе №1 (семестр 2). Жадные алгоритмы. Динамическое программирование №2

В данной лабораторной работе изучается еще один мощный инструмент, жадные алгоритмы. Динамическое программирование тоже продолжаем изучать. Аналогично предыдущим лабораторным работам, есть **два** способа ее выполнения и защиты, однако присутствуют изменения:

- 1 **Базовый уровень.** Решается 5 задач по вариантам. Варианты в табличке внизу, номер вашего варианта соответствует вашему номеру в списке группы. **Посмотреть свой номер нужно, например, в [журнале успеваемости по дисциплине](#).** Вариант в течении семестра менять нельзя!

Условия базового уровня.

- **Всего баллов.** За базовый уровень максимум за защиту и отчет **в сумме** можно получить 9 баллов, если сдаете в срок (1 месяц, до 23 марта 2022 года включительно).
- **Замена.** *Одну* задачу можно заменить без потерь **внутри** уровня сложности, или на более сложную. На уровень ниже - нельзя заменить.
- **Сложность.** У каждой задачи подписано количество баллов, которые можно за нее получить, и по этим баллам задачи распределены на 4 уровня сложности:
 - * 1 уровень, задачи по 0,5 баллов: с 1 по 7;
 - * 2 уровень, задачи по 1 баллу: с 8 по 12;
 - * 3 уровень, задачи по 1.5 - 2 балла: с 13 по 16;
 - * 4 уровень, задачи по 2.5 - 3 балла: с 17 по 21.
 - * Задача №22 не входит ни в один уровень и ни в один вариант.
- **Что это означает?** Это означает, например, что задачу №21 можно заменить на №17 или 18, т.к. они внутри одного уровня.
- **Замена на задачу ниже уровнем.** Вы можете заменить одну задачу на уровень ниже, но тогда вы потеряете разницу в стоимости между ними.

- Замена второй задачи влечет за собой штраф в дополнительные 0.5 балла.
- **Отчет.** На защиту *без отчета*, оформленного по правилам и загруженного в гитхаб, можно *не приходить*!
- Правила оформления отчета:
 - * Формат файла - pdf;
 - * Титульный лист с названием учреждения и факультета, в котором вы учитесь, курса, названия работы, а также ваши ФИО, группа; ФИО преподавателя; город, дата выполнения;
 - * Описание задания к задаче;
 - * Описание вашего решения и исходный код;
 - * Описание проведенных тестов;
 - * Выводы по каждой задаче и в целом по проделанной работе.
- Хорошо оформленный отчет может добавить до 1 балла к защите, на усмотрение преподавателя. Т.е. можно получить аж 10 баллов в итоге!
- За плохо оформленный отчет может сниматься до 2 баллов, также по усмотрению преподавателя. Т.е. можно получить только 7 баллов даже с учетом правильности всех задач.
- **Дедлайн.** Если вы сдаете лабораторную работу *позже* срока (начиная с 24.03.2022), то суммарная базовая стоимость понижается на 2 балла, т.е. до 7 баллов. Эти 2 балла можно компенсировать решением дополнительных задач.

2 Продвинутый уровень. Решаются 5 задач по вариантам, причем принципы, описанные выше для базового случая тоже учитываются. До максимальных 15 баллов вы можете набрать, решая другие задачи не из вашего варианта, выбирая произвольно с учетом стоимости задач. На ваше усмотрение.

Например, решив свой вариант №1, и сделав отчет, вы предполагаете, что получите 9 баллов. Далее вы можете решить еще, например, 19 и 20 задачу, которые стоят по 3 балла каждая. Или, например, 13 и 18 задачи дадут + 4 балла к итоговой сумме.

Условие на дедлайн так же действует для продвинутого уровня (-2 балла).

P.s. Решать все задачи вообще не обязательно. Но если хочется, то каждая решенная задача 4 уровня сложности (в т.ч. 22), которая превышает лимит в максимальные 15 баллов, может быть дополнительно учтена по количеству баллов в конце семестра в счет, например, экзамена.

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

1 задача. Максимальная стоимость добычи (0.5 балла)

Вор находит гораздо больше добычи, чем может поместиться в его сумку. Помогите ему найти самую ценную комбинацию предметов, предполагая, что любая часть предмета добычи может быть помещена в его сумку.

Цель - реализовать алгоритм для задачи о дробном рюкзаке.

- **Формат ввода / входного файла (input.txt).** В первой строке входных данных задано целое число n - количество предметов, и W - вместимость сумки. Следующие n строк определяют значения веса и стоимости предметов. В i -ой строке содержатся целые числа p_i и w_i - стоимость и вес i -го предмета, соответственно.
- **Ограничения на входные данные.** $1 \leq n \leq 10^3$, $0 \leq W \leq 2 \cdot 10^6$, $0 \leq p_i \leq 2 \cdot 10^6$, $0 \leq w_i \leq 2 \cdot 10^6$ для всех $1 \leq i \leq n$. Все числа - целые.
- **Формат вывода / выходного файла (output.txt).** Выведите максимальное значение стоимости долей предметов, которые помещаются в сумку. Абсолютная погрешность между ответом вашей программы и оптимальным значением должно быть не более 10^{-3} . Для этого выведите свой ответ как минимум с четырьмя знаками после запятой (иначе ваш ответ, хотя и будет рассчитан правильно, может оказаться неверным из-за проблем с округлением).
- Ограничение по времени. 2 сек.
- Примеры:

input.txt	output.txt
3 50	180.0000
60 20	
100 50	
120 30	

Чтобы получить значение 180, берем первый предмет и третий предмет в сумку.

input.txt	output.txt
1 10	166.6667
500 30	

Здесь просто берем одну треть единственного доступного предмета.

2 задача. Заправки (0.5 балла)

Вы собираетесь поехать в другой город, расположенный в d км от вашего родного города. Ваш автомобиль может проехать не более m км на полном баке, и вы начинаете с полным баком. По пути есть заправочные станции на расстояниях $stop_1, stop_2, \dots, stop_n$ из вашего родного города. Какое минимальное количество заправок необходимо?

- **Формат ввода / входного файла (input.txt).** В первой строке содержится d - целое число. Во второй строке - целое число m . В третьей строке находится количество заправок на пути - n . И, наконец, в последней строке - целые числа через пробел - остановки $stop_1, stop_2, \dots, stop_n$.
- **Ограничения на входные данные.** $1 \leq d \leq 10^5$, $1 \leq m \leq 400$, $1 \leq n \leq 300$, $1 < stop_1 < stop_2 < \dots < stop_n < d$
- **Формат вывода / выходного файла (output.txt).** Предполагая, что расстояние между городами составляет d км, автомобиль может проехать не более m км на полном баке, а заправки есть на расстояниях $stop_1, stop_2, \dots, stop_n$ по пути, *выведите минимально необходимое количество заправок*. Предположим, что машина начинает ехать с полным баком. Если до места назначения добраться невозможно, выведите -1 .
- Ограничение по времени. 2 сек.
- Примеры:

input.txt	output.txt
950 400 4 200 375 550 750	2

В первом примере расстояние между городами 950 км, на полном баке машина может проехать максимум 400 км. Достаточно сделать две заправки: в точках 375 и 750. Это минимальное количество заправок, так как при одной заправке можно проехать не более 800 км.

input.txt	output.txt	input.txt	output.txt
10 3 4 1 2 5 9	-1	200 250 2 100 150	0

Во втором примере до заправки в точке 9 добраться нельзя, так как предыдущая заправка слишком далеко.

В последнем примере нет необходимости заправлять бак, так как автомобиль стартует с полным баком и может проехать 250 км, а расстояние до пункта назначения составляет 200 км.

3 задача. Максимальный доход от рекламы (0.5 балла)

У вас есть n объявлений для размещения на популярной интернет-странице. Для каждого объявления вы знаете, сколько рекламодатель готов платить за один клик по этому объявлению. Вы настроили n слотов на своей странице и оценили ожидаемое количество кликов в день для каждого слота. Теперь ваша цель - распределить рекламу по слотам, чтобы максимизировать общий доход.

- **Постановка задачи.** Даны две последовательности a_1, a_2, \dots, a_n (a_i - прибыль за клик по i -му объявлению) и b_1, b_2, \dots, b_n (b_i - среднее количество кликов в день i -го слота), нужно разбить их на n пар (a_i, b_j) так, чтобы сумма их произведений была максимальной.
- **Формат ввода / входного файла (input.txt).** В первой строке содержится целое число n , во второй - последовательность целых чисел a_1, a_2, \dots, a_n , в третьей - последовательность целых чисел b_1, b_2, \dots, b_n .
- **Ограничения на входные данные.** $1 \leq n \leq 10^3$, $-10^5 \leq a_i, b_i \leq 10^5$, для всех $1 \leq i \leq n$.
- **Формат вывода / выходного файла (output.txt).** Выведите максимальное значение $\sum_{i=1}^n a_i c_i$, где c_1, c_2, \dots, c_n является перестановкой b_1, b_2, \dots, b_n .
- Ограничение по времени. 2 сек.
- Примеры:

input.txt	output.txt	input.txt	output.txt
1	897	3	23
23		1 3 -5	
39		-2 4 1	

Во втором примере $23 = 3 \cdot 4 + 1 \cdot 1 + (-5) \cdot (-2)$.

4 задача. Сбор подписей (0.5 балла)

Вы несете ответственность за сбор подписей всех жильцов определенного здания. Для каждого жильца вы знаете период времени, когда он или она находится дома. Вы хотите собрать все подписи, посетив здание как можно меньше раз.

Математическая модель этой задачи следующая. Вам дан набор отрезков на прямой, и ваша цель - отметить как можно меньше точек на прямой так, чтобы каждый отрезок содержал хотя бы одну отмеченную точку.

- **Постановка задачи.** Дан набор из n отрезков $[a_0, b_0], [a_1, b_1], \dots, [a_{n-1}, b_{n-1}]$ с координатами на прямой, найдите минимальное количество m точек такое, чтобы каждый отрезок содержал хотя бы одну точку. То есть найдите набор целых чисел X минимального размера такой, чтобы для любого отрезка $[a_i, b_i]$ существовала точка $x \in X$ такая, что $a_i \leq x \leq b_i$.
- **Формат ввода / входного файла (input.txt).** Первая строка входных данных содержит количество отрезков n . Каждая из следующих n строк содержит два целых числа a_i и b_i (через пробел), определяющие координаты концов i -го отрезка.
- **Ограничения на входные данные.** $1 \leq n \leq 10^2$, $0 \leq a_i, b_i \leq 10^9$ - целые для всех $1 \leq i \leq n$.
- **Формат вывода / выходного файла (output.txt).** Выведите минимальное количество m точек в первой строке и целочисленные координаты этих m точек (через пробел) во второй строке. Вывести точки можно в любом порядке. Если таких наборов точек несколько, можно вывести любой набор. (Нетрудно видеть, что всегда существует множество точек минимального размера, для которых все координаты точек - целые числа.)
- Ограничение по времени. 2 сек.
- Примеры:

№	input.txt	output.txt	№	input.txt	output.txt
1	3 1 3 2 5 3 6	1 3	2	4 4 7 1 3 2 5 5 6	2 3 6

В первом примере у нас есть три отрезка: $[1, 3]$, $[2, 5]$, $[3, 6]$ (длиной 2, 3, 3 соответственно). Все они содержат точку с координатой 3: $1 \leq 3 \leq 3$, $2 \leq 3 \leq 5$, $3 \leq 3 \leq 6$.

Во втором примере, второй и третий отрезки содержат точку с координатой 3, а первый и четвертый отрезки содержат точку с координатой 6. Все четыре отрезка не могут быть покрыты одной точкой, так как отрезки $[1, 3]$ и $[5, 6]$ не пересекаются.

5 задача. Максимальное количество призов (0.5 балла)

Вы организуете веселый конкурс для детей. В качестве призового фонда у вас есть n конфет. Вы хотели бы использовать эти конфеты для раздачи k лучшим местам в конкурсе с естественным ограничением, заключающимся в том, что чем выше место, тем больше конфет. Чтобы осчастливить как можно больше детей, вам нужно найти наибольшее значение k , для которого это возможно.

- **Постановка задачи.** Необходимо представить заданное натуральное число n в виде суммы как можно большего числа попарно различных натуральных чисел. То есть найти максимальное k такое, что n можно записать как $a_1 + a_2 + \dots + a_k$, где a_1, \dots, a_k - натуральные числа и $a_i \neq a_j$ для всех $1 \leq i < j \leq k$.
- **Формат ввода / входного файла (input.txt).** Входные данные состоят из одного целого числа n .
- **Ограничения на входные данные.** $1 \leq n \leq 10^9$.
- **Формат вывода / выходного файла (output.txt).** В первой строке выведите максимальное число k такое, что n можно представить в виде суммы k попарно различных натуральных чисел. Во второй строке выведите эти k попарно различных натуральных чисел, которые в сумме дают n (если таких представлений много, выведите любое из них).
- Ограничение по времени. 2 сек.
- Примеры:

№	input.txt	output.txt	№	input.txt	output.txt
1	6	3 1 2 3	2	8	3 1 2 5

№	input.txt	output.txt
3	2	1 2

6 задача. Максимальная зарплата (0.5 балла)

В качестве последнего вопроса успешного собеседования ваш начальник дает вам несколько листов бумаги с цифрами и просит составить из этих цифр наибольшее число. Полученное число будет вашей зарплатой, поэтому вы очень заинтересованы в максимизации этого числа. Как вы можете это сделать?

На лекциях мы рассмотрели следующий алгоритм составления наибольшего числа из заданных *однозначных* чисел.

```
1 def LargestNumber(Digits):
2     answer = ''
3     while Digits:
4         maxDigit = float('-inf')
5         for digit in Digits:
6             if digit >= maxDigit:
7                 maxDigit = digit
8         answer += str(maxDigit)
9         Digits.remove(maxDigit)
10    return answer
```

К сожалению, этот алгоритм работает только в том случае, если вход состоит из однозначных чисел. Например, для ввода, состоящего из двух целых чисел 23 и 3 (23 не однозначное число!) возвращается 233, в то время как наибольшее число на самом деле равно 323. Другими словами, использование наибольшего числа из входных данных в качестве первого числа *не является безопасным ходом*.

Ваша цель в этой задаче – настроить описанный выше алгоритм так, чтобы он работал не только с однозначными числами, но и с произвольными положительными целыми числами.

- **Постановка задачи.** Составить наибольшее число из набора целых чисел.
- **Формат ввода / входного файла (input.txt).** Первая строка входных данных содержит целое число n . Во второй строке даны целые числа a_1, a_2, \dots, a_n .
- **Ограничения на входные данные.** $1 \leq n \leq 10^2$, $1 \leq a_i \leq 10^3$ для всех $1 \leq i \leq n$.
- **Формат вывода / выходного файла (output.txt).** Выведите наибольшее число, которое можно составить из a_1, a_2, \dots, a_n .
- Ограничение по времени. 2 сек.
- Пример:

input.txt	output.txt	input.txt	output.txt
2	221	3	923923
21 2		23 39 92	

7 задача. Проблема сапожника (0.5 балла)

- **Постановка задачи.** В некоей воинской части есть сапожник. Рабочий день сапожника длится K минут. Заведующий складом оценивает работу сапожника по количеству починенной обуви, независимо от того, насколько сложный ремонт требовался в каждом случае. Дано n сапог, нуждающихся в починке. Определите, какое максимальное количество из них сапожник сможет починить за один рабочий день.
- **Формат ввода / входного файла (input.txt).** В первой строке вводятся натуральные числа K и n . Затем во второй строке идет n натуральных чисел t_1, \dots, t_n - количество минут, которые требуются, чтобы починить i -й сапог.
- **Ограничения на входные данные.** $1 \leq K \leq 1000$, $1 \leq n \leq 500$, $1 \leq t_i \leq 100$ для всех $1 \leq i \leq n$
- **Формат вывода / выходного файла (output.txt).** Выведите одно число – максимальное количество сапог, которые можно починить за один рабочий день.
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 256 мб.
- Примеры:

input.txt	output.txt
10 3 6 2 8	2

input.txt	output.txt
3 2 10 20	0

8 задача. Расписание лекций (1 балл)

- **Постановка задачи.** Вы наверно знаете, что в ИТМО лекции читают одни из лучших преподаватели мира. К сожалению, лекционных аудиторий у нас не так уж и много, особенно на Биржевой, поэтому каждый преподаватель составил список лекций, которые он хочет прочитать студентам. Чтобы студенты, в начале февраля, увидели расписание лекций, необходимо его составить прямо сейчас. И без вас нам здесь не справиться. У нас есть список заявок от преподавателей на лекции для одной из аудиторий. Каждая заявка представлена в виде временного интервала $[s_i, f_i)$ - время начала и конца лекции. Лекция считается открытым интервалом, то есть какая-то лекция может начаться в момент окончания другой, без перерыва. Необходимо выбрать из этих заявок такое подмножество, чтобы суммарно выполнить максимальное количество заявок. Учтите, что одновременно в лекционной аудитории, конечно же, может читаться лишь одна лекция.
- **Формат ввода / входного файла (input.txt).** В первой строке вводится натуральное число N - общее количество заявок на лекции. Затем вводится N строк с описаниями заявок - по два числа в каждом s_i и f_i для каждой лекции i . Гарантируется, что $s_i < f_i$. Время начала и окончания лекции - натуральные числа, не превышают 1440 (в минутах с начала суток).
- **Ограничения на входные данные.** $1 \leq N \leq 1000, 1 \leq s_i, f_i \leq 1440$
- **Формат вывода / выходного файла (output.txt).** Выведите одно число – максимальное количество заявок на проведение лекций, которые можно выполнить.
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 256 мб.
- Примеры:

input.txt	output.txt
1	1
5 10	

input.txt	output.txt
3	2
1 5	
2 3	
3 4	

9 задача. Распечатка (1 балл)

- **Постановка задачи.** Диссертация дело сложное, особенно когда нужно ее печатать. При этом вам нужно распечатать не только текст самой диссертации, так и другие материалы (задание, рецензии, отзывы, афторефераты для защиты и т.п.). Вы оценили объём печати в N листов. Фирма, готовая размножить печатные материалы, предлагает следующие финансовые условия. Один лист она печатает за A_1 рублей, 10 листов - за A_2 рублей, 100 листов - за A_3 рублей, 1000 листов - за A_4 рублей, 10000 листов - за A_5 рублей, 100000 листов - за A_6 рублей, 1000000 листов - за A_7 рублей. При этом не гарантируется, что один лист в более крупном заказе обойдется дешевле, чем в более мелком. И даже может оказаться, что для любой партии будет выгодно воспользоваться тарифом для одного листа. Печать конкретного заказа производится или путем комбинации нескольких тарифов, или путем заказа более крупной партии. Например, 980 листов можно распечатать, заказав печать 9 партий по 100 листов плюс 8 партий по 10 листов, сделав 98 заказов по 10 листов, 980 заказов по 1 листу или заказав печать 1000 (или даже 10000 и более) листов, если это окажется выгоднее. Требуется по заданному объему заказа в листах N определить минимальную сумму денег в рублях, которой будет достаточно для выполнения заказа.
- **Формат ввода / входного файла (input.txt).** На вход программе сначала подается число N – количество листов в заказе. В следующих 7 строках ввода находятся натуральные числа $A_1, A_2, A_3, A_4, A_5, A_6, A_7$ соответственно.
- **Ограничения на входные данные.** $1 \leq N \leq 2 \times 10^9, 1 \leq A_i \leq 10^6$
- **Формат вывода / выходного файла (output.txt).** Выведите одно число – минимальную сумму денег в рублях, которая нужна для выполнения заказа. Гарантируется, что правильный ответ не будет превышать 2×10^9 .
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 256 мб.
- Примеры:

input.txt	output.txt	input.txt	output.txt
980	882	980	900
1		1	
9		10	
90		100	
900		1000	
1000		900	
10000		10000	
10000		10000	

10 задача. Яблоки (1 балл)

- **Постановка задачи.** Алисе в стране чудес попались n волшебных яблок. Про каждое яблоко известно, что после того, как его съешь, твой рост сначала уменьшится на a_i сантиметров, а потом увеличится на b_i сантиметров. Алиса очень голодная и хочет съесть все n яблок, но боится, что в какой-то момент ее рост s станет равным нулю или еще меньше, и она пропадет совсем. Помогите ей узнать, можно ли съесть яблоки в таком порядке, чтобы в любой момент времени рост Алисы был больше нуля.
- **Формат ввода / входного файла (input.txt).** В первой строке вводятся натуральные числа n и s – число яблок и начальный рост Алисы. В следующих n строках вводятся пары натуральных чисел a_i, b_i через пробел.
- **Ограничения на входные данные.** $1 \leq n \leq 1000$, $1 \leq s \leq 1000$, $1 \leq a_i, b_i \leq 1000$.
- **Формат вывода / выходного файла (output.txt).** Если яблоки съесть нельзя, выведите число -1. Иначе выведите n чисел – номера яблок, в том порядке, в котором их нужно есть.
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 256 мб.
- Примеры:

input.txt	output.txt	input.txt	output.txt
3 5	1 3 2	3 5	-1
2 3		2 3	
10 5		10 5	
5 10		5 6	

11 задача. Максимальное количество золота (1 балл)

Вам дается набор золотых слитков, и ваша цель - набрать как можно больше золота в свою сумку. Существует только одна копия каждого слитка, и для каждого слитка вы можете либо взять его, либо нет (т.е. вы не можете взять часть слитка).

- **Постановка задачи.** Даны n золотых слитков, найдите максимальный вес золота, который поместится в сумку вместимостью W .
- **Формат ввода / входного файла (input.txt).** Первая строка входных данных содержит вместимость W сумки и количество n золотых слитков. В следующей строке записано n целых чисел w_0, w_1, \dots, w_{n-1} , определяющие вес золотых слитков.
- **Ограничения на входные данные.** $1 \leq W \leq 10^4$, $1 \leq n \leq 300$, $0 \leq w_0, \dots, w_{n-1} \leq 10^5$
- **Формат вывода / выходного файла (output.txt).** Выведите максимальный вес золота, который поместится в сумку вместимости W .
- Ограничение по времени. 5 сек.
- Пример:

input.txt	output.txt
10 3	9
1 4 8	

Здесь сумма весов первого и последнего слитка равна 9.

- Обратите внимание, что в этой задаче все предметы имеют одинаковую стоимость на единицу веса по простой причине: все они сделаны из золота.

12 задача. Последовательность (1 балл)

- **Постановка задачи.** Дана последовательность натуральных чисел a_1, a_2, \dots, a_n , и известно, что $a_i \leq i$ для любого $1 \leq i \leq n$. Требуется определить, можно ли разбить элементы последовательности на две части таким образом, что сумма элементов в каждой из частей будет равна половине суммы всех элементов последовательности.
- **Формат ввода / входного файла (input.txt).** В первой строке входного файла находится одно целое число n . Во второй строке находится n целых чисел a_1, a_2, \dots, a_n .
- **Ограничения на входные данные.** $1 \leq n \leq 40000, 1 \leq a_i \leq i$.
- **Формат вывода / выходного файла (output.txt).** В первую строку выходного файла выведите количество элементов последовательности в любой из получившихся двух частей, а во вторую строку через пробел номера этих элементов. Если построить такое разбиение невозможно, выведите -1.
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 256 мб.
- Примеры:

input.txt	output.txt
3	1
1 2 3	3

13 задача. Сувениры (1.5 балла)

Вы и двое ваших друзей только что вернулись домой после посещения разных стран. Теперь вы хотели бы поровну разделить все сувениры, которые все трое накупили.

- **Формат ввода / входного файла (input.txt).** В первой строке дано целое число n . Во второй строке даны целые числа v_1, v_2, \dots, v_n , разделенные пробелами.
- **Ограничения на входные данные.** $1 \leq n \leq 20$, $1 \leq v_i \leq 30$ для всех i .
- **Формат вывода / выходного файла (output.txt).** Выведите 1, если можно разбить v_1, v_2, \dots, v_n на три подмножества с одинаковыми суммами и 0 в противном случае.
- Ограничение по времени. 5 сек.
- Примеры:

input.txt	output.txt	input.txt	output.txt
4 3 3 3 3	0	1 40	0

input.txt	output.txt
11 17 59 34 57 17 23 67 1 18 2 59	1

Здесь $34 + 67 + 17 = 23 + 59 + 1 + 17 + 18 = 59 + 2 + 57$.

input.txt	output.txt
13 1 2 3 4 5 5 7 7 8 10 12 19 25	1

Здесь $1 + 3 + 7 + 25 = 2 + 4 + 5 + 7 + 8 + 10 = 5 + 12 + 19$.

14 задача. Максимальное значение арифметического выражения (2 балла)

В этой задаче ваша цель - добавить скобки к заданному арифметическому выражению, чтобы максимизировать его значение.

$$\max(5 - 8 + 7 \times 4 - 8 + 9) = ?$$

- **Постановка задачи.** Найдите максимальное значение арифметического выражения, указав порядок применения его арифметических операций с помощью дополнительных скобок.
- **Формат ввода / входного файла (input.txt).** Единственная строка входных данных содержит строку s длины $2n + 1$ для некоторого n с символами s_0, s_1, \dots, s_{2n} . Каждый символ в четной позиции s является цифрой (то есть целым числом от 0 до 9), а каждый символ в нечетной позиции является одной из трех операций из $+, -, *$
- **Ограничения на входные данные.** $0 \leq n \leq 14$ (следовательно, строка содержит не более 29 символов).
- **Формат вывода / выходного файла (output.txt).** Выведите максимально возможное значение заданного арифметического выражения среди различных порядков применения арифметических операций.
- Ограничение по времени. 5 сек.
- Пример:

input.txt	output.txt	input.txt	output.txt
1+5	6	5-8+7*4-8+9	200

Здесь $200 = (5 - ((8 + 7) * (4 - (8 + 9))))$.

15 задача. Удаление скобок (2 балла)

- **Постановка задачи.** Дана строка, составленная из круглых, квадратных и фигурных скобок. Определите, какое наименьшее количество символов необходимо удалить из этой строки, чтобы оставшиеся символы образовывали правильную скобочную последовательность.
- **Формат ввода / входного файла (input.txt).** Во входном файле записана строка, состоящая из s символов: круглых, квадратных и фигурных скобок $()$, $[]$, $\{\}$. Длина строки не превосходит 100 символов.
- **Ограничения на входные данные.** $1 \leq s \leq 100$.
- **Формат вывода / выходного файла (output.txt).** Выведите строку максимальной длины, являющейся правильной скобочной последовательностью, которую можно получить из исходной строки удалением некоторых символов.
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 256 мб.
- Пример:

input.txt	output.txt
([])	[]

16 задача. Продавец (2 балла)

- **Постановка задачи.** Продавец техники хочет объехать n городов, посетив каждый из них ровно один раз. Помогите ему найти кратчайший путь.
- **Формат ввода / входного файла (input.txt).** Первая строка входного файла содержит натуральное число n – количество городов. Следующие n строк содержат по n чисел – длины путей между городами. В i -й строке j -е число – $a_{i,j}$ – это расстояние между городами i и j .
- **Ограничения на входные данные.** $1 \leq n \leq 13$, $0 \leq a_{i,j} \leq 10^6$, $a_{i,j} = a_{j,i}$, $a_{i,i} = 0$.
- **Формат вывода / выходного файла (output.txt).** В первой строке выходного файла выведите длину кратчайшего пути. Во второй строке выведите чисел – порядок, в котором нужно посетить города.
- Ограничение по времени. 1 сек.
- Ограничение по памяти. 256 мб.
- Пример:

input.txt
5
0 183 163 173 181
183 0 165 172 171
163 165 0 189 302
173 172 189 0 167
181 171 302 167 0
output.txt
666
4 5 2 3 1

17 задача. Ход конем (2.5 балла)

- **Постановка задачи.** Шахматная ассоциация решила оснастить всех своих сотрудников такими телефонными номерами, которые бы набирались на кнопочном телефоне ходом коня. Например, ходом коня набирается телефон 340-49-27. При этом телефонный номер не может начинаться ни с цифры 0, ни с цифры 8.

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

Напишите программу, определяющую количество телефонных номеров длины N , набираемых ходом коня. Поскольку таких номеров может быть очень много, выведите ответ по модулю 10^9 .

- **Формат ввода / входного файла (input.txt).** Во входном файле записано одно целое число N .
- **Ограничения на входные данные.** $1 \leq N \leq 1000$.
- **Формат вывода / выходного файла (output.txt).** Выведите в выходной файл искомое количество телефонных номеров по модулю 10^9 .
- Ограничение по времени. 1 сек.
- Ограничение по памяти. 256 мб.
- Примеры:

input.txt	output.txt	input.txt	output.txt
1	8	2	16

18 задача. Кафе (2.5 балла)

- **Постановка задачи.** Около университета недавно открылось новое кафе, в котором действует следующая система скидок: при каждой покупке более чем на 100 рублей покупатель получает купон, дающий право на один бесплатный обед (при покупке на сумму 100 рублей и меньше такой купон покупатель не получает). Однажды вам на глаза попался прейскурант на ближайшие n дней. Внимательно его изучив, вы решили, что будете обедать в этом кафе все n дней, причем каждый день вы будете покупать в кафе ровно один обед. Однако стипендия у вас небольшая, и поэтому вы хотите по максимуму использовать предоставляемую систему скидок так, чтобы ваши суммарные затраты были минимальны. Требуется найти минимально возможную суммарную стоимость обедов и номера дней, в которые вам следует воспользоваться купонами.
- **Формат ввода / входного файла (input.txt).** В первой строке входного файла дается целое число n - количество дней. В каждой из последующих n строк дано одно неотрицательное целое число s_i - стоимость обеда в рублях на соответствующий день i .
- **Ограничения на входные данные.** $0 \leq n \leq 100$, $0 \leq s_i \leq 300$ для всех $0 \leq i \leq n$.
- **Формат вывода / выходного файла (output.txt).** В первой строке выдайте минимальную возможную суммарную стоимость обедов. Во второй строке выдайте два числа k_1 и k_2 - количество купонов, которые останутся у вас неиспользованными после этих n дней и количество использованных вами купонов соответственно. В последующих k_2 строках выдайте в возрастающем порядке номера дней, когда вам следует воспользоваться купонами. Если существует несколько решений с минимальной суммарной стоимостью, то выдайте то из них, в котором значение k_1 максимально (на случай, если вы когда-нибудь ещё решите заглянуть в это кафе). Если таких решений несколько, выведите любое из них.
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 64 мб.
- Пример:

input.txt	output.txt	input.txt	output.txt
5	260	3	220
110	0 2	110	1 1
40	3	110	2
120	5	110	
110			
60			

19 задача. Произведение матриц (3 балла)

- **Постановка задачи.** В произведении последовательности матриц полностью расставлены скобки, если выполняется один из следующих пунктов:

- Произведение состоит из одной матрицы.
- Оно является заключенным в скобки произведением двух произведений с полностью расставленными скобками.

Полная расстановка скобок называется оптимальной, если количество операций, требуемых для вычисления произведения, минимально.

Требуется найти оптимальную расстановку скобок в произведении последовательности матриц.

- **Формат ввода / входного файла (input.txt).** В первой строке входных данных содержится целое число n - количество матриц. В n следующих строк содержится по два целых числа a_i и b_i - количество строк и столбцов в i -той матрице соответственно. Гарантируется, что $b_i = a_{i+1}$ для любого $1 \leq i \leq n - 1$.
- **Ограничения на входные данные.** $1 \leq n \leq 400$, $1 \leq a_i, b_i \leq 100$ для всех $1 \leq i \leq n$.
- **Формат вывода / выходного файла (output.txt).** Выведите оптимальную расстановку скобок. Если таких расстановок несколько, выведите любую.
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 64 мб.
- Пример:

input.txt	output.txt
3	((AA)A)
10 50	
50 90	
90 20	

- В данном примере возможно всего две расстановки скобок: $((AA)A)$ и $(A(AA))$. При первой количество операций будет равно $10 \cdot 50 \cdot 90 + 10 \cdot 90 \cdot 20 = 63000$, а при второй - $10 \cdot 50 \cdot 20 + 50 \cdot 90 \cdot 20 = 100000$.

20 задача. Почти палиндром (3 балла)

- **Постановка задачи.** Слово называется палиндромом, если его первая буква совпадает с последней, вторая – с предпоследней и т.д. Например: «abba», «madam», «x».

Для заданного числа K слово называется почти палиндромом, если в нем можно изменить не более K любых букв так, чтобы получился палиндром. Например, при $K = 2$ слова «reactor», «kolobok», «madam» являются почти палиндромами (подчеркнуты буквы, заменой которых можно получить палиндром).

Подсловом данного слова являются все слова, получающиеся путем вычеркивания из данного нескольких (возможно, одной или нуля) первых букв и нескольких последних. Например, подсловами слова «cat» являются слова «с», «а», «t», «са», «ат» и само слово «cat» (а «ct» подсловом слова «cat» не является).

Требуется для данного числа K определить, сколько подслов данного слова S являются почти палиндромами.

- **Формат входного файла (input.txt).** В первой строке входного файла вводятся два натуральных числа: N – длина слова и K . Во второй строке записано слово S , состоящее из N строчных английских букв.
- **Ограничения на входные данные.** $1 \leq N \leq 5000$, $0 \leq K \leq N$.
- **Формат выходного файла (output.txt).** В выходной файл требуется вывести одно число – количество подслов слова S , являющихся почти палиндромами (для данного K).
- Ограничение по времени. 2 сек.
- Ограничение по памяти. 256 мб.
- Примеры:

input.txt	output.txt	input.txt	output.txt
5 1	12	3 3	6
abcde		aaa	

- Проверить можно по [ссылке](#).

21 задача. Игра в дурака (3 балла)

- **Постановка задачи.** Петя очень любит программировать. Недавно он решил реализовать популярную карточную игру «Дурак». Но у Пети пока маловато опыта, ему срочно нужна Ваша помощь.

Как известно, в «Дурака» играют колодой из 36 карт. В Петиной программе каждая карта представляется в виде строки из двух символов, где первый символ означает ранг ('6', '7', '8', '9', 'T', 'J', 'Q', 'K', 'A') карты, а второй символ означает масть ('S', 'C', 'D', 'H'). **Ранги перечислены в порядке возрастания старшинства.**

Пете необходимо решить следующую задачу: сможет ли игрок, обладая набором из N карт, отбить M карт, которыми под него сделан ход? Для того чтобы отбиться, игроку нужно покрыть каждую из карт, которыми под него сделан ход, картой из своей колоды. Карту можно покрыть либо старшей картой той же масти, либо картой козырной масти. Если кроющаяся карта сама является козырной, то её можно покрыть только старшим козырем. Одной картой можно покрыть только одну карту.

- **Формат входного файла (input.txt).** В первой строке входного файла находятся два натуральных числа N и M , а также символ R , означающий козырную масть. Во второй строке перечислены N карт, находящихся на руках у игрока. В третьей строке перечислены M карт, которые необходимо отбить. Все карты отделены друг от друга одним пробелом.
- **Ограничения на входные данные.** $N \leq 35$, $M \leq 4$, $M \leq N$.
- **Формат выходного файла (output.txt).** В выходной файл выведите «YES» в случае, если отбиться можно, либо «NO», если нельзя.
- Ограничение по времени. 1 сек.
- Ограничение по памяти. 16 мб.
- Примеры:

input.txt	output.txt	input.txt	output.txt
6 2 C KD KC AD 7C AH 9C 6D 6C	YES	4 1 D 9S KC AH 7D 8D	NO

- Проверить можно по [ссылке](#).

22 задача. Симпатичные узоры (4 балла)

- **Постановка задачи.** Компания BrokenTiles планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер 1×1 метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника $M \times N$ метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезных проблемы: во первых, каждый новый клиент очевидно захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во вторых, этот узор должен быть симпатичным. Как показало исследование, узор является **симпатичным**, если в нем нигде не встречается квадрата 2×2 метра, полностью покрытого плитками одного цвета.

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

- **Формат входного файла (input.txt).** В первой строке входного файла находятся два положительных целых числа, разделенные пробелом – M и N .
- **Ограничения на входные данные.** $1 \leq N \times M \leq 30$.
- **Формат выходного файла (output.txt).** Выведите в выходной файл единственное число – количество различных симпатичных узоров, которые можно выложить во дворе размера $M \times N$. Узоры, получающиеся друг из друга сдвигом, поворотом или отражением считаются различными.
- Ограничение по времени. 1 сек.
- Ограничение по памяти. 16 мб.
- Примеры:

input.txt	output.txt	input.txt	output.txt
2 2	14	3 3	322

- Проверить можно по [ссылке](#).