1 Сортировки

1.1. Даны два отсортированных по неубыванию массива a и b. Определите, есть ли в них одинаковые числа. Время $O(n)$.

1.2. Даны два отсортированных по неубыванию массива a и b. Найдите такие i и j, что разница $|a_i - b_j|$ минимальна. Время $O(n)$.

1.3. Даны два отсортированных по неубыванию массива a и b и число S. Найдите такие i и j, что сумма $a_i + b_j = S$. Время $O(n)$.

1.4. Даны два отсортированных по неубыванию массива a и b. Найдите число пар (i, j), таких, что $a_i = b_j$. Время $O(n)$.

1.5. Даны два отсортированных по неубыванию массива a и b. Найдите число пар (i, j), таких, что $a_i > b_j$. Время $O(n)$.

1.6. Дан массива a. Пара (i, j), такая, что $i < j$ и $a_i > a_j$ называется <u>инверсией</u>. Пусть в массиве длины n ровно k инверсий. Докажите, что сортировка вставками работает за $O(n + k)$.

1.7. Дан массива a. Найдите число инверсий в нем. Время $O(n \log n)$

1.8. Покажите, что при правильной (какой?) реализации сортировка слиянием является устойчивой (то есть, не меняет порядок равных элементов).

1.9. Покажите, как сделать сортировку слиянием снизу вверх, без рекурсии.

2 Рекуррентные соотношения

2.1. Докажите по индукции, что если $T(n) = 2T(n/2 + 20) + n$, то $T(n) = O(n \log n)$.

2.2. Докажите по индукции, что если $T(n) = 2T(n/2 + \log n) + n$, то $T(n) = O(n \log n)$.

2.3. Докажите по индукции, что если $T(n) = \log n \cdot T(n/\log n) + n$, то $T(n) = O(n \log n)$.

2.4. Докажите по индукции, что если $T(n) = 2T(n/2) + n$, то $T(n) = \Omega(n \log n)$ (оценка снизу).

2.5. Докажите по индукции, что если $T(n) = 2T(\sqrt{n}) + 1$, то $T(n) = O(\log n)$.

2.6. Пусть $T(n) = aT(n/b) + n^k$.

(2.6.1) Покажите, что если $k < \log_b a$, то $T(n) = O(n^{\log_b a})$.
(2.6.2) Покажите, что если $k > \log_b a$, то $T(n) = O(n^k)$.
(2.6.3) Покажите, что если $k = \log_b a$, то $T(n) = O(n^k \log n)$.

2.7. Для каждой из приведенных программ и функций оцените время ее работы a)

```
for i in range(n):
  j = 0
  while j * j < i:
    j += 1
```

b)

```
for i in range(n):
  j = i
  while j > 0:
    j = j // 2
```

c)

```
def f(n):
  if n == 0:
    return 1
  else
    return 5 * f(n // 3)
```

d)

```
def f(n):
  if n == 0:
    return 1
  else
    return f(n // 3) + f(n // 3)
```

2.8. Пусть время работы алгоритма А описывается соотношением $T_A(n) = 7T_A(n/2) + n^2$, а время работы алгоритма В описывается соотношением $T_B(n) = aT_B(n/4) + n$. При каких значениях a второй алгоритм работает асимптотически быстрее первого?

3 Куча, сортировка кучей

3.1. Пусть в куче лежат числа от 1 до 1000, по одному разу каждое. Какое минимальное число может лежать в куче на самом нижнем уровне?

3.2. Пусть в двоичной куче лежит n элементов. Сколько листьев у соответствующего дерева?

3.3. Пусть в двоичной куче лежит n элементов. Покажите, что в дереве не более $\lceil \frac{n}{2^{h+1}} \rceil$ поддеревьев высоты h (высота поддерева — это расстояние до самого далекого листа в поддереве).

3.4. Пусть в куче лежат числа от 1 до n, по одному разу каждое. В каком случае операция `removeMin` будет работать минимальное, а в каком — максимальное время?

3.5. Пусть дерево кучи организовано таким образом, что у каждой вершины (кроме нижнего слоя) не два ребенка, а три. Какие номера будут у детей вершины i в этом случае?

3.6. Пусть дерево кучи организовано таким образом, что у каждой вершины (кроме нижнего слоя) d детей (при этом d — не константа, а параметр). За какое время работают основные операции над кучей в этом случае? Приведите оценку зависимости от n и d.

3.7. Как добавить в кучу операцию изменения ключа элемента?

3.8. Как из двух куч сделать структуру данных, которая одновременно может искать и удалять как максимум, так и минимум.

3.9. Постройте структуру данных, которая одновременно может искать и удалять как максимум, так и минимум, но при этом тратит столько же памяти, сколько обычная куча (одна, а не две). (Указание: хранить на четных слоях максимальный элемент, а на нечетных — минимальный).

3.10. На базе куч постройте структуру данных, которая может находить и удалять медиану ($n/2$ элемент в отсортированном порядке).

3.11. Есть k отсортированных массивов, содержащих в сумме n элементов. Слейте их в один отсортированный массив за время $O(n \log k)$.

3.12. Приведите пример массива, в котором все элементы различны, и сортировка кучей работает за $O(n)$.

3.13. Можно ли добавить в кучу k элементов быстрее, чем за $O(k \log n)$?.

3.14. Петя хотел построить кучу за $O(n)$, но сделал это не совсем верно:

```
for i = 0 .. n - 1:
    sift_down(i)
```

Покажите, что такой алгоритм иногда не работает.

4 Быстрая сортировка, K-я порядковая статистика, прочие задачи на сортировку

4.1. Приведите пример, когда быстрая сортировка работает за $\Omega(n^2)$, если в качестве разделителя выбирается: a) самый левый элемент отрезка, b) самый правый элемент отрезка, c) центральный элемент отрезка (`a[(l + r) / 2]`).

4.2. То же самое, но в качестве разделителя выбирается медиана из этих трех элементов.

4.3. Представьте, что злоумышленник знает, по какому алгоритму выбирается разделяющий элемент (например, если он знает, какой у вас генератор случайных чисел). Как он может составить тест, на котором быстрая сортировка работает за $\Omega(n^2)$?

4.4. Предложите способ, который бы защищал быструю сортировку от плохих тестов, но при этом не ухудшал время ее работы на случайных тестах. То есть, пусть быстрая сортировка работает за время $A \cdot n \log n$. Предложите модификацию, которая в среднем тоже работает примерно за $A \cdot n \log n$, а в худшем — за $B \cdot n \log n$.

4.5. Сколько дополнительной памяти использует быстрая сортировка в среднем и в худшем случае?

4.6. Как модифицировать быструю сортировку, чтобы в худшем случае требовалось $O(\log n)$ дополнительной памяти?

4.7. У вас есть n болтов и n подходящих к ним гаек, все болты (и, соответственно, все гайки) разного диаметра. Посмотрев на два болта (или две гайки), сложно понять, какой больше, а какой меньше, поэтому единственная операция, которая у вас есть — взяв какой-то болт и какую-то гайку, сравнить их диаметры (если не пролезает — значит болт больше, если болтается — значит меньше). Найдите для каждого болта подходящую гайку за $O(n \log n)$.

4.8. Есть массив. Нужно получить первые k элементов массива в отсортированном порядке. За какое минимальное время это можно сделать?

4.9. Что будет, если в алгоритме Блюма, Флойда, Пратта, Ривеста и Тарьяна заменить константу 5 на 3 или 7?

4.10. В отсортированном массиве размера n изменили k элементов (неизвестно, каких именно). Отсортируйте полученный массив за $O(n + k \log k)$.

4.11. Как с помощью генератора случайных чисел получить случайную перестановку? Нужно, чтобы каждая перестановка появлялась с вероятностью $1/n!$.

4.12. Дан массив из n чисел. Необходимо для каждого элемента найти число элементов, которые меньше его. Время работы $O(n \log n)$.

4.13. Дано два массива: a и b. Найдите такие i и j, что $a_i < a_j$ и $b_i < b_j$, или скажите, что найти невозможно. Время работы $O(n \log n)$.

4.14. Дан массив a из n чисел, постройте массив b, состоящий из натуральных чисел не больше n, такой, что $a_i < a_j$ тогда и только тогда, когда $b_i < b_j$. Время работы $O(n \log n)$.

4.15. В ряд стоят n ящиков. Необходимо отсортировать их по номерам. У вас есть кран, который умеет делать одну команду: `swap(i, j)`, которая меняет местами ящики i и j. Постройте план работы для крана, который отсортирует ящики, совершив минимальное число операций. Время работы (вашей, не крана) $O(n \log n)$.

4.16. Покажите, что любую сортировку можно сделать устойчивой, не поменяв асимптотику работы и потратив $O(n)$ дополнительной памяти.

5 Цифровая сортировка + разные задачи на сортировку

5.1. Дан массив из n чисел от 1 до k, разработайте структуру данных, которая за $O(1)$ отвечает на запросы вида «Сколько в массиве элементов от a до b?». Время на предподсчет $O(n + k)$.

5.2. Как с помощью цифровой сортировки сортировать строки (напрмер, состоящие только из латинских букв) в лексикографическом порядке за $O(\sum len(s_i)$?

5.3. Возьмем массив a из n элементов, каждый из которых — это число 1 до n. Циклический сдвиг номер i — это последовательность $a_i, a_{i+1}, ..., a_{n-1}, a_0, a_1, ..., a_{i-1}$. Предложите алгоритм сортировки циклических сдвигов в лексикографическом порядке за время $O(n^2)$.

5.4. Предложите алгоритм сортировки циклических сдвигов за $O(n \log n)$. Указание: зная порядок на подстроках длины L порядок на подстроках длины $2L$ можно восстановить за $O(n)$.

5.5. Пусть известно, что массив длины n из чисел от 1 до n получен с помощью генератора случайных чисел, каждое число независимо получено с помощью равномерного распределения. Предложите модификацию алгоритма сортировки подсчетом, который сортирует данный массив за $O(n)$, используя лишь $O(\sqrt{n})$ дополнительной памяти (обе оценки должны выполняться в среднем).

5.6. Есть массив, состоящий из n натуральных чисел. Найти минимальное натуральное число, которого нет в массиве, за $O(n)$.

5.7. Есть массив длины n. Посчитайте, сколько есть способов упорядочить его элементы в неубывающем порядке.

5.8. На прямой живут n друзей, i-й друг живет в точке x_i. Они хотят встретиться в одной точке. Помогите им найти такую точку, чтобы суммарное расстояние, которое они пройдут, было бы минимально.

5.9. На прямой живут n друзей, i-й друг живет в точке x_i. Они хотят встретиться в одной точке. Помогите им найти такую точку, чтобы сумма квадратов расстояний, которое они пройдут, была бы минимальна.

5.10. Даны два массива a и b, состоящих из целых чисел. Проверьте, можно ли в массиве a выбрать k чисел, а в массиве b — m чисел так, что любое число, выбранное в первом массиве, строго меньше любого числа, выбранного во втором массиве.

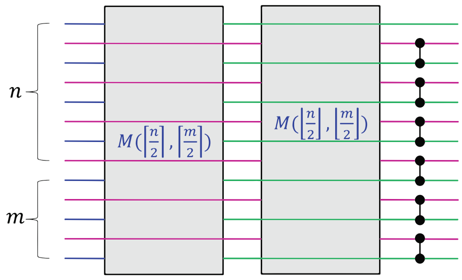
5.11. Есть улица длины l, которая освещается n фонарями, i-й фонарь находится в точке a_i. Фонарь освещает все точки улицы, которые находятся от него на расстоянии не больше d, где d — некоторое положительное число, общее для всех фонарей. Найдите минимальное d, при котором вся улица освещена.

5.12. Есть массив, состоящий из n натуральных чисел. Можно уменьшать числа, но чтобы они оставались натуральными. Какое максимальное число различных чисел может быть в массиве после нескольких таких операций?

5.13. Дан массив из $2n$ различных элементов, на которых определен линейный порядок (то есть их можно сравнивать). Нужно разбить их на n пар, так, чтобы отрезки, границами которых являются числа из пар, не пересекались (напрмер $(1, 2)$, $(7, 10)$, $(4, 6)$). Можно ли решить эту задачу быстрее чем за $O(n \log n)$?

5.14. Пусть мы хотим показать, что некоторая сортирующая сеть сливает два отсортированных массива в один. Покажите, что для этого достаточно показать, что она сливает массивы, состоящие из 0 и 1.

5.15. Будем строить сортирующую сеть $M(n, m)$ для слияния двух отсортированных массивов размера n и m следующим образом. Разделим элементы обоих массивов на четные и нечетные, рекурсивно сольем отдельно четные, отдельно нечетные, в конце добавим компараторы для элементов $2k$ и $2k + 1$ (см картинку):



(5.15.1) Покажите, что такая сеть действительно сливает два отсортированных массива.

(5.15.2) Каково будет общее число компараторов в такой сети?

(5.15.3) Какова будет глубина такой сети?

6 Двоичный поиск

6.1. Дан массив положительных чисел, отвечать на запросы: «Какое максимальное число элементов из начала массива можно взять, чтобы их сумма была не больше X?».

6.2. Задан массив, полученный циклическим сдвигом из отсортированного по возрастанию. Все элементы массива различны. Требуется за $O(\log n)$ найти в нем заданный элемент.

6.3. Пусть в предыдущей задаче убрали условие, что все элементы массива различны. Можно ли в таком массиве найти заданный элемент за $O(\log n)$?

6.4. Задан массив, полученный приписыванием одного отсортированного по возрастанию массива в конец другому отсортированному по возрастанию. Все элементы массива различны. Требуется за $O(\log n)$ найти в нем заданный элемент.

6.5. Задан массив, полученный приписыванием отсортированного по убыванию массива в конец отсортированному по возрастанию. Все элементы массива различны. Требуется за $O(\log n)$ найти в нем заданный элемент.

6.6. Задан массив, полученный приписыванием отсортированного по убыванию массива в конец отсортированному по возрастанию и затем циклическим сдвигом получившегося массива. Все элементы массива различны. Требуется за $O(\log n)$ найти в нем заданный элемент.

6.7. Пусть выполняется целочисленный двоичный поиск с начальными значениями L, R. Предложите алгоритм определения за $O(\log n)$ по заданным значениям l и r, могут ли они возникнуть в процессе двоичного поиска.

6.8. Есть n кучек предметов, в i-ой кучке a_i предметов. Все предметы пронумерованы сквозной нумерацией, так, что в кучке с меньшим номером номера предметов меньше. За $O(\log n)$ отвечать на запрос: в какой кучке находится предмет номер x?

6.9. В игре есть n типов ресурсов, для постройки одного юнита требуется a_i единиц ресурса i для всех i от 1 до n. У Пети есть b_i единиц ресурса i и еще d единиц золота. Одну единицу золота можно обменять на d_i единиц ресурса i. Сколько юнитов может построить Петя?

6.10. В выборах участвуют n кандидатов. По последним опросам, за кандидата i готовы проголосовать a_i избирателей. Вы хотите, чтобы ваш кандидат победил (набрал больше голосов, чем любой другой кандидат). За s рублей можно изменить мнение одного избирателя. Сколько надо потратить денег на такую избирательную компанию?

6.11. Покажите, что если в условии на функцию в троичном поиске заменить строгое убывание и возрастание на нестрогое, то найти минимум быстро нельзя.

6.12. Пусть фунция $f(x)$ имеет вид $f(x) = \sum |a_i x + b_i|$ для некоторых (неизвестных) a_i и b_i. Можно ли искать минимум этой функции троичным поиском?

6.13. Есть отсортированный массив из n чисел. Нужно выбрать из них k так, чтобы минимальная разница между соседними выбранными числами была как можно больше.

6.14. Есть массив из n чисел. Нужно выбрать в нем k пар чисел (каждое число может быть только в одной паре), так, чтобы максимальная разность чисел в паре была как можно меньше.

6.15. То же самое, но нужно выбрать k наборов по d чисел.

6.16. Домик черепашки расположен в точке 0 числовой прямой. Однажды она узнала, что скоро вырастут n одуванчиков, одуванчик i вырастет в точке x_i во время t_i, причем все x_i и t_i положительные, а так же $x_{i+1} > x_i$ и $t_{i+1} > t_i$. Черепашка выходит из дома в момент 0, она движется со скоростью не больше v. Чтобы съесть одуванчик, ей нужно остановиться около него на время d. За какое время она сможет съесть все одуванчики и вернуться домой?

6.17. Даны n кенгуру с сумками. У каждого кенгуру есть размер (целое число). Кенгуру может поместиться в сумке другого кенгуру тогда и только тогда, когда размер кенгуру-носителя как минимум в два раза больше размера кенгуру-пассажира. Каждый кенгуру может нести не более одного кенгуру, а кенгуру-пассажир не может носить никаких кенгуру. Кенгуру-пассажира не видно, когда он в сумке кенгуру-носителя. Пожалуйста, разработайте такой план рассадки кенгуру, чтобы было видно как можно меньше кенгуру.

7 Стеки и очереди

7.1. Добавьте в стек и очередь операцию `getSum()`, возвращающую сумму элементов в стеке/очереди. Время работы $O(1)$. Дополнительная память $O(1)$.

7.2. Добавьте в стек операцию `getMin()`, возвращающую минимальный элемент в стеке. Время работы $O(1)$. Дополнительная память $O(size)$ ($size$ — число элементов в стеке).

7.3. Используя стек, научитесь вычислять выражения в постфиксной записи (это когда оператор ставится после аргументов, например, выражение `4-((1+2)*3)` в постфиксной записи выглядит так: `4 1 2 + 3 * -`.

7.4. Научитесь вычислять выражения в инфиксной записи со скобками (обычные выражения). Для простоты можно считать, что в выражении проставлены все скобки (то есть внутри скобок вычисляется только один оператор, например так можно: $(4 - ((1 + 2) * 3))$, а так — нет: $(1 + 2 + 3)$.

7.5. Научитесь по выражению в инфиксной записи строить выражение в постфиксной записи.

7.6. Был массив чисел, все соседние числа были различны. С этим массивом несколько раз проделали операцию: вставить в массив два одинаковых числа рядом. По конечному состоянию массива восстановите его исходное состояние.

7.7. Дан массив из целых чисел. Для каждого элемента найдите ближайший элемент слева, меньший его.

8 Амортизационный анализ

8.1. Проанализировать работу вектора, если расширение происходит в A раз $(1 < A)$. Какое значение A лучше выбирать на практике?

8.2. Пусть выделение массива памяти любого размера происходит за время $O(1)$. Разработайте вектор с истинной (не амортизированной) стоимостью всех операций $O(1)$ и памятью $O(n)$. (Подсказка: надо распределить операцию копирования массива на несколько операций).

8.3. Добавьте в очередь операцию `getMin()`, возвращающую минимальный элемент в очереди. Амортизированное время работы $O(1)$.

8.4. Реализуйте дек на трех стеках с амортизированным временем работы всех операций $O(1)$.

8.5. Добавьте в стек операцию `clear`, удаляющую все элементы. Амортизированная стоимость операции должна быть $O(1)$.

8.6. Битовый счетчик хранит число в виде массива двоичных цифр. Изначально все цифры равны 0. Операция `inc` увеличивает счетчик на 1. Реализуйте операцию `inc` и докажите, что амортизированное время ее работы $O(1)$.

8.7. Как сделать счетчик, который совершает операцию `inc` за настоящее $O(1)$? (погуглите счетчик Кнута).

8.8. Мультистек — это последовательность стеков, максимальный размер i-го стека равен 3^i. Изначально все стеки пустые. Новый элемент добавляется в первый стек. Если он уже заполнен, то все его элементы перекладываются во второй, если он также заполнен, то его элементы перекладываются в третий, и т. д. Покажите, что амортизированное время добавления элемента в мультистек $O(\log n)$.

8.9. Покажите, как сделать, чтобы в мультистеке стеки были всегда отсортированы (при этом амортизированное время добавления элемента все еще $O(\log n)$).

8.10. Покажите, как в полученной структуре найти и удалить минимальный элемент за $O(\log n)$). (В результате должна получиться структура данных на базе стеков, в которую можно добавлять и удалять минимум за $O(\log n)$).

8.11. Как сделать мультистек с настоящим временем работы $O(\log n)$?

9 Связные списки

9.1. Как развернуть односвязный список за время $O(n)$ с $O(1)$ дополнительной памяти?

9.2. Для экономии памяти в двусвязных списках иногда вместо двух указателей хранят только их битовый XOR. (например, если `prev[i] = 5`, `next[i] = 3`, то вместо них храним `prevnext[i] = 6`). Как работать с таким списком?

9.3. Дан набор из n элементов, в каждом есть ссылка на какой-то другой. Проверьте, правда ли эти элементы образуют один большой кольцевой список (менять ссылки нельзя). (Время $O(n)$, память $O(1)$)

9.4. Дан набор из n элементов, в каждом есть ссылка на какой-то другой. Проверьте, правда ли эти элементы образуют один большой линейный список (начальный элемент неизвестен, менять ссылки нельзя). (Время $O(n)$, память $O(1)$).

9.5. Дан набор из n элементов, в каждом есть ссылка на следующий и предыдущий. Проверьте, правда ли эти элементы образуют несколько кольцевых списков (менять ссылки нельзя). (Время $O(n)$, память $O(1)$)

9.6. Дан набор из n элементов, в каждом есть ссылка на следующий и предыдущий. Проверьте, правда ли эти элементы образуют несколько связных списков и, если да, сконкатенируйте эти списки в один большой (в любом порядке). (Время $O(n)$, память $O(1)$)

9.7. Слить два отсортированных односвязных списка в один за время $O(n)$ с $O(1)$ дополнительной памяти.

9.8. Отсортировать связный список за время $O(n \log n)$ с $O(1)$ дополнительной памяти.

9.9. Придумайте, как хранить связный список, чтобы его можно было развернуть за $O(1)$ (с сохранением всех остальных операций).

9.10. Есть таблица $n \times n$. С ней проводят m операций следующего вида: вырезать прямоугольный кусок, развернуть его на 180° и вклеить на то же место. Выведите состояние таблицы после всех операций. Время $O(nm)$.

9.11. Есть полоска бумаги, разбитая на n одинаковых квадратных клеток. На клетках написаны числа от 1 до n. С полоской совершают m действий следующего вида: отсчитывают x_i клеток от левого конца полоски и сгибают полоску в этом месте, кладя правую половину поверх левой. Выведите финальное состояние полоски после всех сгибов. Время $O(m + n)$.

9.12. Покажите как реализовать двоичную кучу в Pointer Machine Model.

9.13. Покажите как сделать в Pointer Machine Model структуру, которая позволяет делать двоичный поиск в списке (список отсортирован и не меняется, требуется отвечать на запросы: найти максимальный элемент не больше x).

10 Система непересекающихся множеств

10.1. Добавьте в СНМ операции `getMin(x)`, `getMax(x)`, `getSize(x)`, возвращающие минимум, максимум и число элементов в множестве x.

В следующих задачах используются термины из теории графов, которые мы пока не проходили. Но вы не пугайтесь, они все несложные, если не знаете какой-то из них — загуглите.

10.2. В изначально пустой граф добавляются ребра одно за другим. После каждого добавления найдите размер самой большой компоненты связности (по числу вершин).

10.3. Есть пустой граф. Делают два вида запросов: 1) добавить ребро 2) найти число ребер в компоненте связности x.

10.4. Есть пустой граф. Делают два вида запросов: 1) добавить ребро 2) найти число компонент связности, являющихся деревьями.

10.5. Есть пустой граф. Делают два вида запросов: 1) добавить ребро 2) найти число компонент связности, являющихся циклами.

10.6. Есть пустой граф. Делают два вида запросов: 1) добавить ребро 2) найти число компонент связности, являющихся двудольными графами.

10.7. Пусть у всех элементов в СНМ есть вес. Добавьте в СНМ операции: 1) увеличить все веса в заданном множестве на d, 2) найти текущий вес элемента x.

10.8. Дан массив a, заполненный нулями, Делают два вида запросов: 1) $a_i := 1$ 2) найти число непрерывных отрезков из единиц.

10.9. Дан массив a, заполненный нулями, Делают два вида запросов: 1) $a_i := 1$ 2) найти ближайший к i ноль.

10.10. Дано дерево a, все вершины покрашены в черный цвет, Делают два вида запросов: 1) покрасить заданную вершину в белый цвет 2) найти ближайшего черного предка данной вершины.

10.11. Дан массив a из положительных чисел. Найдите отрезок с максимальным значением произведения (сумма $\times$ минимум).

10.12. Дан массив a из положительных чисел. Для каждого числа найдите, для скольки разных отрезков оно является минимумом.

10.13. Докажите амотризированную оценку $O(\log n)$ для эвристики сжатия путей (без других эвристик).

10.14. Для кластеризации изображений решают следующую задачу. Есть картинка $n \times m$ пикселей (для простоты будем считать, что каждый пиксель задается одним числом). Мы хотим выделить на ней части примерно одинакового цвета. Для этого нужно разбить картинку на k связных областей так, чтобы минимальная разность значений пикселей на границе областей была максимально возможной.

11 Кучи

11.1. Будем сливать обычные (двоичные) кучи следующим способом: при слиянии брать кучу меньшего размера и перекладывать ее элементы во второую кучу. Пусть у нас было n куч из одного элемента, и мы в некотором порядке слили их в одну. Сколько времени в худшем случае на это уйдет?

11.2. Вспомните, как добавить в двоичную кучу k элементов асимптотически быстрее, чем $O(k \log n)$. Может это поможет улучшить время в предыдущей задаче?

11.3. Сколько вершин ранга r в биномиальном дереве ранга k?

11.4. Разработайте алгоритм слияния биномиальных куч, который идет от больших деревьев к меньшим, с тем же временем работы.

11.5. В биномиальную кучу размера n добавляется один элемент. Как быстро понять, сколько операций слияния биномиальных деревьев произойдет?

11.6. Покажите, что амортизированная стоимость операции `insert` равна $O(1)$ (при том, что амортизированная стоимость остальных операций $O(\log n)$).

11.7. Как можно модифицировать биномиальную кучу, чтобы `insert` выполнялось за истинное $O(1)$? (вспомните счетчик Кнута)

11.8. Добавьте в биномиальную кучу операцию увеличения ключа, за $O(\log n)$.

11.9. Можно ли сделать операцию увеличения ключа быстрее, чем $O(\log n)$?

11.10. Докажите, что максимальный ранг дерева в фибоначчиевой куче $O(\log n)$.

11.11. Рассмотрим следующие деревья. Деревья F_1 и F_2 состоят из одной вершины. Дерево F_k состоит из дерева F_{k-1}, к корню которого подвешено дерево F_{k-2}. Можно ли на базе таких деревьев построить что-то похожее на биномиальную кучу, с таким же временем работы?

11.12. Покажите, что для любого n существует последовательность операций, приводящая к появлению в фибоначчиевой куче дерева в виде цепочки из n элементов (бамбука).

11.13. Разрешим вершины с двумя звездочками (у которых удалили двух детей). Как изменится алгоритм и время работы?

11.14. В одном месте (в каком?) для реализации фибоначчиевой кучи нам понадобился массив. Попробуйте заменить его на список и реализовать фибоначчиеву кучу в Pointer Machine Model.

12 Динамическое программирование

12.1. Кузнечик прыгает из клетки 1 в клетку n, длина прыжка может быть от 1 до k. Найти число различных путей. Время $O(n)$.

12.2. Кузнечик прыгает из клетки 1 в клетку n, длина прыжка может быть от 1 до k. У каждой клетки есть стоимость. Найти путь с минимальнойть стоимостью. Время $O(n)$.

12.3. Кузнечик прыгает из клетки 1 в клетку n, каждый раз на 1 или 2 клетки вперед. У каждой клетки есть стоимость. Найти число различных путей минимальной стоимости. Время $O(n)$.

12.4. Кузнечик прыгает из клетки 1 в клетку n, каждый раз на 1 или 2 клетки вперед. На каждой клетке написана буква. Найти такой путь, чтобы строка, которую прочитает кузнечик, была лексикографически минимальной. Время $O(n^2)$.

12.5. Черепашка ползет из клетки $(0,0)$ в клетку (n,m). На некоторых клетках растут цветочки. Черепашка соберет все цветочки, которые встретит на пути. Найдите максимальное **нечетное** число цветочков, которое она может собрать (пропускать цветы на пути нельзя). Время $O(nm)$.

12.6. Черепашка ползет из клетки $(0,0)$ в клетку (n,m) и собирает цветочки. Найдите число различных путей, на которых черепашка соберет хотя бы k цветков. Время $O(nmk)$.

12.7. Черепашка ползет из клетки $(0,0)$ в клетку (n,m) и собирает цветочки. На каждой клетке растут красные или синие цветочки. Черепашка хочет пройти таким путем, чтобы число клеток с синими и красными цветочками было одинаковым. Помогите ей найти такой путь.

12.8. Черепашка ползет из клетки $(0,0)$ в клетку (n,m). Для каждой клетки известно число a_i. Если это число положительное, то черепашка получает столько рублей, а если отрицательное — то тратит. Сколько денег нужно иметь черепашке в начале пути, чтобы добраться до конца и всегда по дороге иметь неотрицательное число денег. Время $O(nm)$.

12.9. Задана последовательность чисел. Требуется удалить из нее минимальное число элементов, чтобы она стала возрастающей. Время $O(n^2)$.

12.10. Задана последовательность чисел. Требуется найти число способов удалить из нее некоторые элементы, чтобы она стала возрастающей. Время $O(n^2)$.

12.11. Задана последовательность чисел. Требуется разбить ее на максимальное число отрезков, чтобы сумма чисел в каждом следующем отрезке была больше, чем в предыдущем. Время $O(n^3)$.

12.12. Задано уравнение вида A + B = C, где A, B и C — неотрицательные целые числа длины n, в десятичной записи которых некоторые цифры заменены знаками вопроса. Например: `?2 + 34 = 4?`. Требуется подставить вместо знаков вопроса цифры, чтобы это равенство стало верным.

12.13. Петя и Вася играют в игру: в ряд выложены n карточек, на i-й карточке написано число a_i. За один ход можно взять одну, две или три карты с конца ряда. Игра заканчивается, когда нет больше карт. Выигрывает тот, у кого в конце игры сумма чисел на картах максимальна. Кто выиграет при правильной игре?

12.14. В одном здании крышу поддерживают n металлических балок, стоящих в ряд. Сторож Петрович давно хочет сдать их в металлолом, но он опасается, что если спилить две подряд стоящие балки, то что-нибуть сломается. Вес i-й балки равен a_i. Сколько металлолома может безопасно сдать Петрович?

12.15. У Васи есть калькулятор, который умеет выполнять три операции: прибавить 1, умножить на 2 и умножить на 3. Какое наименьшее число операций необходимо для того, чтобы получить из числа 1 число n.

12.16. В очереди за билетами на концерт Моргенштерна стоит n школьников. Чтобы увеличить скорость движения очереди, два подряд идущих школьника могут договориться, и тот, что идет раньше в очереди купит два билета: на себя и на следующего. Школьник i тратит a_i секунд на покупку одного билета и b_i секунд на покупку двух билетов. За какое минимальное время все смогут купить билеты?

12.17. Вася любит ходить в кино и решать домашку по АиСД. Еще Вася любит разнообразие, поэтому он никогда не делает одно и тоже два дня подряд. Для n последовательных дней известно, есть ли в этот день интересное кино и есть ли в этот день интересная домашка. Какое минимальное число дней Вася будет сидеть без дела?

13 Динамическое программирование - 2

13.1. Как изменится решение задачи о редакционном расстоянии если операции добавления, удаления и изменения имеют разные стоимости?

13.2. Даны две строки a и b. Найти такую строку c, чтобы максимальное из редакционных расстояний: от c до a, и от c до b, было минимально возможным. Время $O(n^2)$.

13.3. Депутаты приняли закон, по которому все имена, которые даются детям, должны соответствовать шаблону, соответствующему региону. Шаблон состоит из букв, а так же символов «?» и «*». Символ «?» можно заменить на одну любую букву, символ «*» — на любую строку (в том числе пустую). Проверьте, что строка соответствует шаблону. Время $O(nm)$ (n и m — длина строки и шаблона).

13.4. Тяжело, когда ~~законы принимают идиоты~~ родители ребенка прописаны в разных регионах, ведь тогда имя ребенка должно соответствовать сразу двум шаблонам. Найдите самое короткое подходящее имя или скажите, что такого нет. Время $O(nm)$ (n и m — длины шаблонов).

13.5. У Пети есть n книг, отсортированных по названию. Книга i имеет высоту h_i. Петя хочет сложить книги в шкаф в отсортированном порядке: первые несколько книг стоят на первой полке, следующие несколько на второй, и т.д. На каждую полку помешается не более m книг. Высота полки равна высоте максимальной книге на полке. Помогите Пете распределить книги по полкам так, чтобы суммарная высота полок была минимальной. Время $O(nm)$

13.6. Та же задача, но время $O(n)$

13.7. Деревня Гадюкино представляет собой n домов, расположенных вдоль прямой дороги. Координата i-го дома a_i. Компания «интернет в каждый дом» хочет поставить в деревне несколько вайфай-точек, покрывающих все дома. Точка с радиусом покрытия r стоит $A + Br^2$ рублей. Найдите минимальную стоимость необходимых точек. Время $O(n^2)$.

13.8. Есть n коров на прямой дороге, известны их координаты. Идет дождь, и нужно укрыть их зонтами. Есть m типов зонтов, i-й зонт стоит c_i рублей и может накрыть коров в радиусе r_i. Сколько денег нужно потратить чтобы укрыть всех коров? Время $O(nm)$.

13.9. Компания «Кака-кола» устроила акцию: нужно собрать крышечки с p звездами в сумме, чтобы получить приз. У Пети есть m крышечек с одной звездой, n с двумя и k с тремя. Сколько призов сможет получить Петя? Время $O(mnk)$.

13.10. Пете нужно посчитать произведение n матриц. Матрица i имеет размер $a_i \times a_{i+1}$. Как вы знаете, чтобы посчитать произведение матриц $a \times b$ и $b \times c$, нужно сделать примерно abc действий. Помогите Пете выполнить все умножения в таком порядке, чтобы суммарное число действий было минимально. Время $O(n^3)$.

13.11. Вам нужно распилить бревно длины L на несколько частей. Точки распила уже отмечены, всего есть n точек, расстояние от левого конца до i-й точки распила a_i. Лесопилка за один раз может сделать один распил, при этом стоимость равна длине бревна, которое нужно распилить. В каком порядке нужно делать распилы, чтобы суммарная стоимость была минимальна? Время $O(n^3)$.

13.12. Деревня Гадюкино представляет собой n домов, расположенных вдоль прямой дороги. Координата i-го дома a_i. В деревне планируется построить k бомбоубежищ. Расставьте их так, чтобы суммарное расстояние от каждого из домов до ближайшего к нему бомбоубежища было минимально. Время $O(n^2k)$.

13.13. Дана строка s длины n. Посчитайте для каждой пары индексов (i, j) длину наибольшего общего префикса строк $s[i..n]$ и $s[j..n]$. Время $O(n^2)$.

13.14. Даны две строки a и b длины n. Можно разбить первую строку на несколько частей, а затем сложить эти части в обратном порядке. Например `ск ле п` $\to$ `п ле ск`. Проверить, можно ли получить строку b из строки a за одну такую операцию. Время $O(n^2)$.

13.15. Есть последовательность из n клеток. Каждую из клеток нужно покрасить в один из m цветов. Стоимость покраски i-й клетки в j-й цвет равна c_{ij}. Нужно покрасить последовательность так, чтобы ее можно было разбить на не более k отрезков, на каждом из которых все клетки одного цвета, за минимальную стоимость. Время $O(nmk)$.

13.16. Даны два w-битных числа A и B. Найти число пар x, y таких, что $x + y = A$ и $x|y = B$ (здесь $|$ — побитовое или). Время $O(w)$.

13.17. Дана последовательность из нулей и единиц длины n. С ней можно производить два типа действий. 1) изменить значение одного элемента на противоположное за A монет, 2) выбрать отрезок и изменить значение всех элементов на нем за B монет. Нужно сделать последовательность из нулей за минимальное число монет. Время $O(n)$.

14 Динамическое программирование - 3

14.1. У мага есть n заклинаний и m единиц маны. Заклининие i тратит c_i маны и наносит врагу урон d_i. Убейте монстра с h здоровьем, потратив как можно меньше маны. Время $O(nm)$.

14.2. Петя и Вася выиграли на олимпиаде n призов, суммарной стоимостью r. Они хотят разделить их между собой так, чтобы разница в суммарной стоимости была минимальна. Время $O(nr)$.

14.3. Та же задача за $O(n \cdot 2^{n/2})$.

14.4. Петя и Вася опять выиграли на олимпиаде $n = 2k$ призов, суммарной стоимостью r. Они хотят разделить их между собой так, чтобы каждому досталось по k призов, а разница в суммарной стоимости была минимальна. Время $O(n^2r)$.

14.5. Та же задача за $O(n \cdot 2^{n/2})$.

14.6. Петя и Вася позвали Колю и выиграли еще одну олимпиаду, опять получив n призов суммарной стоимостью r. Они хотят разделить их на троих так, чтобы разница между максимальной и минимальной долей была минимальна. Время $O(nr^2)$.

14.7. Дан граф, раскрасить его вершины в минимальное число цветов так, чтобы каждое ребро соединяло вершины разного цвета. Время $O(3^n)$.

14.8. Есть набор из n чисел. Разбить его на минимальное число наборов, в каждом из которых НОД всех чисел больше 1. Время $O(3^n)$.

14.9. Есть набор из n чисел. Разбить его на максимальное число наборов, в каждом из которых сумма всех чисел больше или равна k. Время $O(2^n \cdot n)$.

14.10. Коля решает задачу о рюкзаке без стоимостей (нужно набрать максимальный вес не больший S) жадным алгоритмом, он работает так. Пусть сейчас уже взяты некоторые предметы, и их суммарный вес равен T, тогда на следующем шаге алгоритма Коля возьмет предмет максимального веса среди тех, для которых $T + w_i \leq S$. Если таких нет, то алгоритм заканчивает работу. Пусть этот алгоритм получил набор с суммой ANS, а оптимальный ответ равен OPT. На каком тесте отношение ANS/OPT будет минимально?

14.11. У доктора Хауса есть n вариантов, чем может болеть пациент. Он может сделать m разных анализов, анализ i дает положительный результат, если $a_{ij} = 1$, где j — номер болезни пациента. Он делает анализы один за другим, при этом, выбирая, какой следующий анализ сделать, он знает результаты всех предыдущих анализов. За какое минимальное число анализов можно гарантированно поставить диагноз? Время $O(2^n \cdot nm)$.

15 Динамическое программирование - 4

15.1. Дана последовательность, найдите максимальную по длине подпоследовательность из последовательных чисел (например, 5, 6, 7, 8). Время $O(n)$.

15.2. Даны две **перестановки** чисел от 1 до n. Найдите их наибольшую общую возрастающую подпоследовательность (да, одновременно и общую, и возрастающую). Время $O(n^2)$.

15.3. Есть строка. Удалить минимальное число букв, чтобы получился палиндром. Время $O(n^2)$.

15.4. Дан набор из n отрезков (l_i, r_i), нужно выбрать как можно больше отрезков так, чтобы они не пересекались (то есть, для любой пары отрезков $r_i < l_j$ или $r_j < l_i$. Время $O(n^2)$.

15.5. Дана последовательность из n круглых, квадратных и фигурных скобок. Удалите минимальное число скобок, чтобы получилась правильная скобочная последовательность. Время $O(n^3)$.

В следующих заданиях время работы не указано специально, придумайте сами. Чем быстрее получится, тем лучше.

15.6. В одной игре есть n героев, у каждого есть три параметра: сила, ловкость и интеллект, выражающиеся целыми числами от 1 до m. Пете нужно выбрать команду из k героев. Он считает, что для того, чтобы команда была более сбалансированной, нужно, чтобы произведение суммарной силы, суммарной ловкости и суммарного интеллекта было максимально. Помогите Пете составить команду.

15.7. Найти число замощений прямоугольника плашками 3×1?

15.8. Найти число замощений прямоугольника $n \times m$ уголками 2×2 (квадрат без одной клетки).

15.9. Найти число замощений правильного шестиугольника со стороной n ромбами со стороной 1.

15.10. Найти число способов обойти все клетки прямоугольника $n \times m$, посетив каждую клетку ровно один раз и оказавшись в конце в той же клетке, где начал. Перемещаться можно только в соседние клетки по стороне.

15.11. Дана шахматная доска $n \times m$, некоторые клетки которой удалены. Поставить на доску максимальное число королей так, чтобы они не били друг друга.

15.12. Та же задача, но про коней.

15.13. Та же задача, но про слонов.

15.14. Та же задача, но про ладей.

15.15. Та же задача, но про ферзей.

16 Хеш-таблицы

16.1. Пусть число возможных ключей $U > nm$, где n — число элементов в хеш-таблице, а m — ее размер. Покажите, что для любой хеш-функции в худшем случае возможно попадание всех n элементов в одну ячейку хеш-таблицы.

16.2. Пусть $h(k)$ — случайная хеш-функция. Каково математическое ожидание числа коллизий (числа пар $(x, y), x \neq y$, для которых $h(x) = h(y)$)?

16.3. Будем разрешать коллизии с помощью списков, но хранить списки в отсортированном порядке. Как это повлияет на время работы в худшем случае и в среднем?

16.4. Пусть число m — простое. Покажите, что множество хеш-функций $h_a(x) = ax \mod m$ удовлетворяет свойству, что для $x \neq y$, вероятность $p(h(x) = h(y)) = 1/m$.

16.5. Множество хеш-функций называется 2-универсальным, если для любой пары ключей $(x, y), x \neq y$, пара $(h(x), h(y))$ может принимать все m^2 значений равновероятно. Пусть число m — простое.

(16.5.1) Покажите, что множество хеш-функций $h_a(x) = ax \mod m$ не является 2-универсальным.

(16.5.2) Покажите, что множество хеш-функций $h_{ab}(x) = ax + b \mod m$ является 2-универсальным.

16.6. Добавьте в хеш-таблицу возможность пробежаться по всем ее элементам в том порядке, в котором они добавлялись, за $O(n)$.

16.7. Добавьте в хеш-таблицу операцию `merge`, которая объединяет два множества в одно. Амортизированное время работы $O(\log n)$.

16.8. Добавьте в очередь (ну или сразу в дек) операцию `countUnique`, которая возвращает число различных элементов в очереди. Время $O(1)$.

16.9. Есть множество вещественных чисел. Нужно делать две операции за $O(1)$: 1) добавить число, 2) найти какое-нибудь число в диапазоне $(x - \varepsilon, x + \varepsilon)$ (или сказать, что ни одного числа нет) ε одинаковый для всех запросов.

16.10. Посмотрите, как устроен метод `Long.hashCode()` в Java (ссылку не дам, сами загуглите). Как быстро сгенерировать много `Long`ов с одинаковым хешом? Попробуйте положить их в `HashSet<Long>` и затем положить в другой `HashSet<Long>` столько же случайных чисел, посмотрите на сколько различается время работы.

16.11. То же самое со `String.hashCode()`.

16.12. Для того, чтобы избежать коллизий, Петя взял массив очень большого размера. Проблема только в том, что массив заполнен «мусором» — данными, оставленными от прошлого использования. Обнулить все эти данные Петя не может, потому что массив реально огромный. Помогите Пете сделать на базе этого огромного массива хеш-таблицу на n элементов, потратив $O(n)$ дополнительной памяти.

16.13. В хеш-таблице с открытой адресацией при добавлении и поиске элемента x будем изучать элементы начиная с $h_1(x)$ с шагом $h_2(x)$ (то есть значения $(h_1(x) + i \cdot h_2(x)) \mod M$ для $i = 0, 1, \ldots$) для некоторых двух хеш-функций h_1 и h_2. Через сколько шагов такая последовательность зациклится? Как это числ зависит от $h_1(x)$, $h_2(x)$ и M?

16.14. Пусть процессор работает с w-битными числами и поддерживает все стандартные арифметические и битовые операции. Научитесь после предподсчета за $O(w)$ и с $O(w)$ памяти, находить за $O(1)$ времени максимальную степень двойки, на которую делится заданное число.