АиСД у2022. Четвертый семестр
Домашние задания M3234-M3237

⟨Версия от 24 мая 2024 г.⟩

Темы

Неделя 1. Паросочетания в двудольных графах

Устная часть

1.1. Дан ациклический ориентированный граф. Нужно покрыть все его вершины минимальным числом вершинно-непересекающихся путей. Вершина считается покрытой, если она принадлежит какому-то пути. Время работы: $\mathcal{O}(n \cdot (n+m))$.

1.2. Рассмотрим двудольный граф с долями L и R. Найдем в графе максимальное паросочетание. Ориентируем ребра паросочетания из правой доли в левую, а все остальные ребра наоборот: из левой доли в правую. В полученном ориентированном графе запустим от всех не покрытых паросочетанием вершин левой доли обход в глубину. Обозначим за L^+ и R^+ множества вершин левой и правой доли, соответственно, посещенных обходом в глубину. Остальные вершины обозначим за L^- и R^-, соответственно. Таким образом, $L = L^+ \cup L^-$, а $R = R^+ \cup R^-$. Докажите, что в данном ориентированном графе не существует ребер (u, v), таких что:

- ▷ $u \in L^+$, $v \in R^-$,
- ▷ $u \in R^+$, $v \in L^-$,
- ▷ $u \in R^-$, $v \in L^+$.

1.3. Назовем *вершинным покрытием* двудольного графа множество вершин графа VC, такое что для любого ребра (u, v) верно хотя бы одно из двух утверждений: $u \in VC$, $v \in VC$. Докажите, что множество вершин $L^- \cup R^+$ является вершинным покрытием.

1.4. Минимальным вершинным покрытием назовем вершинное покрытие, состоящее из минимального количества вершин. Докажите, что множество из предыдущего задания является минимальным вершинным покрытием.

1.5. *Независимым множеством* в двудольном графе назовем множество вершин IS, такое что для любых двух вершин $u \in IS$ и $v \in IS$ в графе не существует ребра (u, v). Максимальным независимым множеством назовем независимое множество, состоящее из максимального количества вершин. Научитесь искать максимальное независимое множество в двудольном графе за $\mathcal{O}(n \cdot (n+m))$.

1.6. Назовем паросочетание *совершенным*, если оно покрывает все вершины графа. Докажите, что совершенное паросочетание существует в двудольном графе тогда и только тогда, когда для любого подмножества левой доли $A \subseteq L$ верно, что $|A| \le |N(A)|$, где $N(A)$ — множество соседей вершин из множества A.

1.7. Докажите, что если в дереве есть совершенное паросочетание, то оно единственно.

1.8. Докажите, что в d-регулярном двудольном графе существует совершенное паросочетание, и что d-регулярный граф можно разбить на d непересекающихся паросочетаний.

1.9. Есть поле $n \times n$, некоторые клетки которого удалены. Требуется положить на поле максимальное число доминошек 2×1. Доминошки можно поворачивать на 90 градусов. Время работы: $\mathcal{O}(n^3)$.

1.10. Есть поле $n \times n$, некоторые клетки которого удалены. Требуется поставить максимальное число шахматных коней так, чтобы они не били друг друга. Время работы: $\mathcal{O}(n^3)$.

1.11. Есть поле $n \times n$, некоторые клетки которого удалены. Требуется поставить максимальное число шахматных ладей так, чтобы они не били друг друга. Время работы: $\mathcal{O}(n^3)$.

1.12. Есть двудольный граф. Существует паросочетание покрывающее множество $A \subseteq L$, также существует паросочетание покрывающее множество $B \subseteq R$. Докажите, что существует паросочетание покрывающее $A \cup B$.

1.13. Дан двудольный граф на n вершинах и m ребрах. Найдите паросочетание размера хотя бы $\frac{M}{2}$, где M — размер максимального паросочетания. Время работы: $\mathcal{O}(n + m)$.

Неделя 2. Паросочетания

Устная часть

2.14. Почему важно, чтобы увеличивающий путь был простой? Приведите пример, когда существует путь, похожий на увеличивающий (первая и последняя вершины свободные, каждое второе ребро из паросочетания), но не простой (проходит по некоторым ребрам или вершинам несколько раз), но при этом увеличить паросочетание нельзя.

2.15. Почему важно, чтобы у соцветия был стебель? Приведите пример когда есть цикл, похожий на соцветие (нечетной длины, каждое второе ребро из паросочетания), но если его сжать, то в графе потеряется увеличивающий путь (то есть, до сжатия паросочетание можно было увеличить, а после — нельзя).

2.16. Почему нужно разжимать соцветия после нахождения увеличивающего пути? Почему нельзя сжать соцветие навсегда, найти максимальное паросочетание в том, что получилось, а потом разжать обратно?

2.17. Рассмотрим следующий алгоритм поиска увеличивающего пути:

- ▷ Раздвоим вершины: для каждой вершины v создадим в новом графе две вершины v_0 и v_1,
- ▷ Для каждого ребра (u, v) не из паросочетания построим ребро (u_0, v_1) в новом графе,
- ▷ Для каждого ребра (u, v) из паросочетания построим ребро (u_1, v_0) в новом графе,
- ▷ Найдем при помощи обхода в глубину путь из $(s, 0)$ в $(t, 1)$ для некоторой пары свободных вершин s и t.
- ▷ Перенесем этот путь в исходный граф. Легко видеть, что в этом пути каждое второе ребро принадлежит паросочетанию, а значит путь является увеличивающим.

Найдите проблему в этом алгоритме.

2.18. Дан двудольный граф. У каждой вершины левой доли $v \in L$ есть вес $w(v)$ — произвольное положительное число. Максимальным взвешенным паросочетанием назовем такое паросочетание M, для которого сумма весов покрытых вершин левой доли максимальна.

Рассмотрим следующий алгоритм поиска максимального взвешенного паросочетания:

- ▷ Отсортируем вершины левой доли по убыванию веса,
- ▷ Запустим алгоритм Куна от вершин левой доли в порядке сортировки.

Докажите, что данный алгоритм строит максимальное взвешенное паросочетание.

Подсказка: используйте матроиды.

2.19. Каждому ребру заданного графа необходимо сопоставить вещественное число от 0 до 1 так, чтобы для каждой вершины сумма чисел на инцидентных ребрах была не больше 1. Среди всех способов требуется максимизировать сумму чисел. Чем эта задача похожа на паросочетание и почему это не одно и то же?

2.20. Решите предыдущую задачу за полином от размера графа.

2.21. Приведите контрпример для следующего алгоритма поиска размера максимального паросочетания в произвольном графе:

- ▷ Создадим двудольный граф из $2n$ вершин. Для каждой вершины v в исходном графе создадим две вершины в разных долях нового графа: l_v и r_v,
- ▷ Для каждого ребра (u, v) исходного графа создадим в новом графе ребра (l_u, r_v) и (l_v, r_u),
- ▷ Найдем максимальное паросочетание M в двудольном графе,
- ▷ Скажем, что ответ (то есть размер паросочетания в исходном графе) равен $\left\lfloor \frac{|M|}{2} \right\rfloor$.

2.22. Дан двудольный граф. Выберите минимальное число ребер так, чтобы любая вершина была концом ребра из выбранного множества.

2.23. Дан двудольный граф и максимальное паросочетание M в этом графе. Определите, правда ли, что в графе существует единственное минимальное вершинное покрытие. Время работы: $\mathcal{O}(n + m)$.

2.24. Найдите максимальное паросочетание в двудольном графе, если степень каждой вершины левой доли не превосходит 2. Время работы: $\mathcal{O}((n + m) \log (n + m))$.

Неделя 3. Потоки — 1

Устная часть

3.25. Докажите, что если в сети существует поток величины f, то для любого g $(0 \le g < f)$, существует поток величины g.

3.26. Докажите, что если количество ребер в сети равно m, и пропускная способность каждого из них не превосходит $c_{\max}$, то величина максимального потока не превосходит $m \cdot c_{\max}$.

3.27. Сведите задачу о максимальном паросочетании в двудольном графе к нахождению максимального потока. Как связаны разрез и минимальное вершинное покрытие?

3.28. Приведите пример сети с вещественными пропускными способностями, в котором алгоритм Форда-Фалкерсона может никогда не завершиться, но при этом сможет получить сколь угодно близкий к максимальному поток.

3.29. Приведите пример сети с вещественными пропускными способностями, в котором алгоритм Форда-Фалкерсона может никогда не завершиться, и не сможет получить сколь угодно близкий к максимальному поток.

3.30. Научитесь проверять, что в графе существует единственный минимальный разрез.

3.31. Пусть в исходной сети есть ребра (u, v) и (v, u) (возможно, имеющие различные пропускные способности). Как будет работать алгоритм Форда-Фалкерсона в таком случае?

3.32. Как изменить алгоритм Форда-Фалкерсона, чтобы он работал для неориентированного графа? (В такой постановке задачи по ребру (u, v) с пропускной способностью $c(u, v)$ можно пустить не более, чем $c(u, v)$, выбрав направление потока по ребру).

3.33. Рассмотрим следующую модификацию алгоритма Форда-Фалкерсона. Пусть $c_{\max}$ — максимальная пропускная способность среди всех ребер сети. Скажем, что $s = 2^{\lfloor \log_2 c_{\max} \rfloor}$. Алгоритм будет совершать следующие итерации до тех пор, пока $s \ge 1$:

(a) Найти в остаточной сети увеличивающий путь, в котором все остаточные пропускные способности не меньше, чем s.

(b) Если такого пути не существует, уменьшить s в два раза и перейти на следующую итерацию.

(c) В противном случае увеличить поток по найденному пути, перестроить остаточную сеть и перейти к пункту (a).

Докажите, что такой алгоритм корректно решает задачу о максимальном потоке.

3.34. Докажите, что алгоритм, описанный в предыдущем задании, работает за $\mathcal{O}(m^2 \log c_{\max})$.

Неделя 4. Потоки — 2

Устная часть

4.35. Дан неориентированный граф. Найти максимальное число попарно реберно непересекающихся путей из s в t.

4.36. Дан неориентированный граф. Найти максимальное число попарно вершинно непересекающихся путей из s в t.

4.37. Есть поле размера $n \times m$. Некоторые клетки свободны, некоторые заняты горами. На поле есть два замка. Нужно построить в некоторых клетках стены, так чтобы нельзя было пройти от одного замка к другому, при условии, что за один ход можно переместиться в соседнюю по стороне клетку (нельзя ходить по горам и стенам). Какое минимальное число клеток надо застроить?

4.38. Дан неориентированный граф. В некоторых вершинах находится фабрика (всего таких вершин k). Также в некоторых вершинах находятся магазины (всего таких вершин k). Соедините каждую фабрику с магазином при помощи пути так, чтобы никакие два пути не пересекались. То есть фабрики и магазины нужно разбить на пары, и каждую пару соединить путем.

4.39. Докажите реберную теорему Менгера: минимальное число ребер, которые необходимо удалить в графе, чтобы из s в t не было пути, равно максимальному числу реберно непересекающихся путей из s в t.

4.40. Докажите вершинную теорему Менгера: минимальное число вершин, которые необходимо удалить в графе, чтобы из s в t не было пути, равно максимальному числу вершинно непересекающихся путей из s в t (вершины s и t удалять нельзя).

4.41. Постройте граф, в котором $m = \Theta(n^2)$, и алгоритм Диница работает за $\Theta(n^4)$.

4.42. Мальчик Вася не знает алгоритмов поиска максимального потока. Он где-то прочитал про алгоритм Эдмонса-Карпа, но не осознал его. Поэтому когда надо было его реализовать, он сделал ошибку: не добавил обратные ребра. Поэтому при поиске дополняющего пути он всегда идет только вдоль направления ребра в исходной сети. Найдите (по возможности маленький) тест, на котором алгоритм Васи работает неправильно.

4.43. Во сколько раз ответ алгоритма Васи может отличаться от правильного ответа?

4.44. Есть прямоугольный торт размера $n \times m$. На нем лежат k вишенок и k клубничек. Требуется разрезать торт на k связных кусков так, чтобы в каждом куске была и вишенка, и клубничка.

Неделя 5. Потоки — 3

Устная часть

5.45. Дан взвешенный граф G на n вершинах и m ребрах. Веса ребер могут быть отрицательными, но в графе нет отрицательных циклов. Определим для каждой вершины *потенциал* $\varphi(v)$ — некоторое число. Построим новый граф G' на тех же множествах вершин и ребер, что и граф G. Весом ребра (u, v) в графе G' будет величина $w'_{u,v} = w_{u,v} + \varphi(v) - \varphi(u)$, где $w_{u,v}$ — вес ребра (u, v) в графе G. Докажите, что кратчайший путь между вершинами s и t в графе G является кратчайшим путем между вершинами s и t в графе G'.

5.46. Придумайте, как ввести потенциалы $\varphi(v)$ для вершин графа, чтобы веса всех ребер в графе G' были неотрицательными.

5.47. Для каждой пары вершин (s, t) найдите длину кратчайшего пути в графе. Время работы: $\mathcal{O}(nm \log n)$. Обратите внимание, что при небольшом количестве ребер такой алгоритм будет асимптотически лучше, чем алгоритм Флойда, работающий за время $\mathcal{O}(n^3)$.

5.48. Решите задачу Min-Cost-K-Flow за время $\mathcal{O}(nm + k \cdot m \log n)$.

5.49. Дана сеть, в которой у каждого ребра (u, v) помимо пропускной способности имеется параметр $m(u, v)$ — минимальное количество потока, который должен протекать по данному ребру. Найдите произвольный поток, удовлетворяющий данным ограничениям. Время работы: $\mathcal{O}(n^2 \cdot m)$.

5.50. Решите предыдущую задачу при условии, что нужно найти максимальный поток, удовлетворяющий данным ограничениям. Время работы: $\mathcal{O}(n^2 \cdot m)$.

5.51. Найдите в неориентированном графе путь из вершины u в вершину v, проходящий через вершину w. Время работы: $\mathcal{O}(m\sqrt{m})$.

5.52. Дана матрица $n \times n$. Требуется в каждой клетке записать некоторое целое число от 0 до k таким образом, чтобы сумма чисел в i-й строке не превосходила r_i, сумма чисел в j-м столбце не превосходила c_j, а сумма всех чисел была как можно больше. Время работы: $\mathcal{O}(n^4)$.

5.53. Дан полный двудольный граф, каждая доля которого состоит из n вершин. У каждого ребра графа есть вес. Найдите совершенное паросочетание минимального веса. Время работы: $\mathcal{O}(n^4)$.

5.54. Дано поле размера $n \times n$. Каждую клетку поля требуется полить удобрением. Это можно делать тремя способами:

- ▷ Полить клетку (i, j), это будет стоить $a_{i,j}$;
- ▷ Полить все клетки строки i, это будет стоить r_i;
- ▷ Полить все клетки столбца j, это будет стоить c_j.

Требуется полить все клетки хотя бы один раз за минимальную стоимость. Сведите данную задачу к задаче нахождения минимального разреза.

5.55. Пусть сеть представляет собой планарный граф. Для простоты скажем, что вершины заданы точками на плоскости, ребра — отрезками, вершины s и t — это самая левая и самая правая точка, соответственно. Найдите величину максимального потока из s в t быстрее, чем за $\mathcal{O}(nm)$.

5.56. Задан двудольный граф. Нужно раскрасить ребра в k цветов так, чтобы для любой вершины все инцидентные ей ребра были раскрашены в разные цвета. Определите минимальное k, при котором это можно сделать, и найдите раскраску в k цветов, за полиномиальное от размера графа время.

Неделя 6. Задача о назначениях

Устная часть

6.57. Дан взвешенный ориентированный граф. Требуется рокрыть все его вершины простыми непересекающимися циклами минимального суммарного веса.

6.58. Есть множество, на его элементах задан частичный порядок, описанный как некий ориентированный граф. Требуется найти максимальное по размеру множество попарно несравнимых элементов.

6.59. Модифицируйте венгерский алгоритм для поиска по прямоугольной матрице размера $n \times m$ паросочетания размера $\min(n, m)$ (здесь n и m — размеры долей двудольного графа, который можно построить по матрице). Оцените время работы данного алгоритма.

6.60. По заданной квадратной матрице размера $n \times n$ требуется найти такие вектора x и y, что $x_i + y_j > a_{i,j}$ и $\sum x_i + \sum y_i \to \min$.

6.61. Назовем матрицу a *хорошей*, если для любого элемента (i, j) верно, что $a_{i,j} + a_{i+1,j+1} = a_{i+1,j} + a_{i,j+1}$. Дана исходная не обязательно хорошая матрица a. Требуется увеличить некоторые ее элементы таким образом, чтобы она стала хорошей, а суммарное увеличение элементов было как можно меньше.

6.62. Дан взвешенный двудольный граф. Скажем, что вес паросочетания — это вес максимального ребра в нем. Постройте полное паросочетание минимального веса.

6.63. Пусть в графе уже построено полное паросочетание минимального веса, после этого вес одного из ребер увеличивается. Покажите, как перестроить паросочетание (быстрее, чем строить с нуля).

6.64. Пусть в графе уже построено полное паросочетание минимального веса, после этого вес одного из ребер уменьшается. Покажите, как перестроить паросочетание (быстрее, чем строить с нуля).

Неделя 7. Линейное программирование

Устная часть

7.65. Рассмотрим задачу поиска максимума в массиве. Сформулируйте задачу в виде задачи линейного программирования.

7.66. Постройте двойственную задачу к задаче поиска максимума в массиве, попробуйте понять ее физический смысл.

7.67. Рассмотрим задачу поиска вершинного покрытия минимального веса: «Дан граф, каждая вершина имеет вес. Требуется построить вершинное покрытие, в котором суммарный вес выбранных вершин минимален». Сформулируйте задачу в форме задачи линейного программирования.

7.68. Приведите пример графа, на котором в задаче поиска вершинного покрытия минимального веса оптимум достигается не в целой точке.

7.69. Покажите, как построить 2-приближенный алгоритм решения задачи поиска вершинного покрытия минимального веса, используя решение задачи в вещественных числах.

7.70. Рассмотрим задачу поиска кратчайшего пути в графе. Проще всего сформулировать эту задачу в форме задачи линейного программирования, записав ее как поток минимальной стоимости размера 1. Попробуйте это сделать.

7.71. Постройте двойственную задачу к задаче поиска кратчайшего пути, попробуйте понять ее физический смысл.

7.72. Рассмотрим задачу поиска минимального остовного дерева в графе. Рассмотрим некоторое множество из $n-1$ ребра. Попробуем сформулировать свойство, что это множество является остовным деревом. Пустим в середину каждого ребра из истока ребро с пропускной способностью 1, а из каждой вершины в сток — ребро с пропускной способностью $1-\frac{1}{n}$. Покажите, что в такой сети есть поток размера $n-1$ тогда и только тогда, когда данное множество является остовным деревом.

7.73. Сопоставим каждому ребру переменную $x_e \in \{0,1\}$. Используя результат из прошлого пункта, сформулируйте в терминах линейного программирования свойство, что множество ребер, для которых $x_e = 1$, является остовным деревом.

7.74. Рассмотрим задачу о назначениях. Сформулируйте задачу в форме задачи линейного программирования.

7.75. Постройте двойственную задачу к задаче о назначениях, посмотрите на условия на двойственных переменных, попробуйте понять, чему они соответствуют в Венгерском алгоритме.

Неделя 8. Вычислительная геометрия — 1

Устная часть

8.76. При помощи предиката поворота проверьте, что точка P лежит внутри (или на границе) треугольника ABC.

8.77. Проверьте, что точка P лежит внутри выпуклого многоугольника. Время работы: $\mathcal{O}(n)$.

8.78. Даны точки A, O, B и P. При помощи предиката поворота проверьте, что точка P лежит внутри угла AOB.

8.79. Иван заблудился в саванне, и на его поиски выдвинулись n львов. К сожалению, львы ищут Ивана не для того, чтобы помочь ему, а для того чтобы поужинать. В начальный момент времени Иван находится в точке P_0. Он выбирает направление, в котором он будет ехать, спасаясь от львов, и дальше все время едет по прямой. Львы, будучи великолепными охотниками, сразу определяют направление, выбранное Иваном, и планируют его поимку соответственно. В частности, львы понимают, что Иван поедет по прямой. Скорости всех львов постоянны и равны между собой, а также равны скорости передвижения Ивана. Изначально львы находятся в точках $P_1, P_2, \ldots, P_n$. Определите, сможет ли Иван спастись от львов, или ему неминуемо придется стать ужином. В случае, если спастись удастся, определите направление, в котором нужно двигаться Ивану. Считается, что Иван станет ужином, если в какой-то момент времени координаты Ивана и хотя бы одного льва совпадут. Время работы: $\mathcal{O}(n \log n)$.

8.80. В заданном квадрате случайно равновероятно сгенерировали n точек. Докажите, что размер выпуклой оболочки точек — $\mathcal{O}(\log n)$.

8.81. В заданном круге случайно равновероятно сгенерировали n точек. Докажите, что размер выпуклой оболочки точек — $\mathcal{O}(\log n)$.

8.82. Дан выпуклый многоугольник, состоящий из n вершин. Поступают запросы, каждый из которых характеризуется прямой $a_i x + b_i y + c_i = 0$. Научитесь проверять, пересекается ли данная прямая с многоугольником. Время работы на запрос: $\mathcal{O}(\log n)$.

8.83. Даны n точек $P_1, P_2, \ldots, P_n$. Поступают запросы, каждый из которых характеризуется прямой $a_i x + b_i y + c_i = 0$. Научитесь находить точку P_i, расстояние от которой до прямой максимально. Время работы на запрос: $\mathcal{O}(\log n)$.

8.84. Даны два выпуклых многоугольника с n и m вершинами, соответственно. Проверьте, что многоугольники пересекаются за $\mathcal{O}(nm)$.

8.85. Найдите расстояние между двумя выпуклыми многоугольниками за $\mathcal{O}(nm)$.

Неделя 9. Вычислительная геометрия — 2

Устная часть

9.86. Докажите, что сумма Минковского двух выпуклых многоугольников — это выпуклый многоугольник.

9.87. Найдите расстояние между двумя выпуклыми многоугольниками за $\mathcal{O}(n+m)$.

9.88. Дан выпуклый многоугольник. Требуется отвечать на запросы: «дана точка P, необходимо найти касательные к многоугольнику, проходящие через данную точку». Время работы: $\mathcal{O}(\log n)$ на запрос.

9.89. Дан выпуклый многоугольник. Найдите длину максимального отрезка, концы которого лежат на границе многоугольника. Время работы: $\mathcal{O}(n)$.

9.90. Многоугольник называется *звездным*, если существует точка O, такая что любой отрезок, соединяющий точку O и точку на границе многоугольника, целиком лежит внутри многоугольника. Решите задачу о принадлежности точки звездному многоугольнику за $\mathcal{O}(\log n)$ на запрос. Обратите внимание, что выпуклый многоугольник является частным случаем звездного многоугольника.

9.91. Дан произвольный многоугольник. Постройте его триангуляцию за $\mathcal{O}(n^2)$. Триангуляцией называется разбиение многоугольника на $n-2$ треугольника, вершинами которых являются вершины исходного многоугольника, при условии, что все треугольники лежат внутри заданного многоугольника.

9.92. Докажите, что для любых трех хороших (связных, непрерывных, замкнутых, бла-бла-бла), но не обязательно выпуклых, множеств S_1, S_2 и S_3 верно следующее свойство: $S_1 + (S_2 \cup S_3) = (S_1 + S_2) \cup (S_1 + S_3)$, где операция $+$ — это сумма Минковского.

9.93. Как выглядит сумма Минковского двух невыпуклых многоугольников?

9.94. Даны m выпуклых многоугольников с размерами $n_1, n_2, \ldots, n_m$. Пусть $n = n_1 + n_2 + \ldots + n_m$. Найдите сумму Минковского всех многоугольников за $\mathcal{O}(n \log n)$.

9.95. Дан выпуклый многоугольник на n вершинах — это стена. Снаружи от него даны m точек — в них расположены фонари. Фонарь светит во все стороны на любое расстояние. Определить длину освещенной части стены. Время работы: $\mathcal{O}(n \log n)$.

9.96. Даны два множества точек. Найдите прямую, которая их разделяет, то есть такую прямую, чтобы все точки одного множества были по одну сторону, а все точки второго — по другую. Время работы: $\mathcal{O}(n \log n)$.

Неделя 10. Теория чисел

Устная часть

Во всех заданиях можно считать, что арифметические и битовые операции с числами выполняются за $\mathcal{O}(1)$.

10.97. Опишите, как выглядят все решения диофантова уравнения $ax + by = c$.

10.98. Опишите, как выглядят все решения уравнения $ax \equiv b \pmod{m}$.

10.99. Даны целые числа a, b, c и m. Найдите количество пар чисел x и y, таких что $ax + by = c$, $|x| \le m$ и $|y| \le m$. Время работы: $\mathcal{O}(\log \max(a, b, c))$.

10.100. Даны целые числа a, b и c. Найдите такие целые числа x и y, что $ax + by = c$ и $|x| + |y|$ минимально. Время работы: $\mathcal{O}(\log \max(a, b, c))$.

10.101. Научитесь проверять, что целое число n имеет ровно три различных делителя, быстрее, чем за $\mathcal{O}(\sqrt{n})$. В данной задаче нельзя использовать алгоритм Полларда и прочие сложные алгоритмы факторизации.

10.102. Функция $\varphi(n)$, рассмотренная на лекции, называется функцией Эйлера. Докажите, что функция Эйлера является *мультипликативной*. Функция называется мультипликативной, если для любых взаимно-простых a и b верно, что $\varphi(a \cdot b) = \varphi(a) \cdot \varphi(b)$.

10.103. Зная факторизацию числа n, научитесь вычислять $\varphi(n)$ за $\mathcal{O}(\log n)$.

10.104. Зная факторизацию числа n, научитесь вычислять количество различных делителей числа n за $\mathcal{O}(\log n)$.

10.105. Зная факторизацию числа n, научитесь вычислять сумму всех делителей числа n за $\mathcal{O}(\log n)$.

10.106. Пусть $n = p \cdot q$, где p и q — простые числа. Докажите, что, зная n и $\varphi(n)$, можно быстро факторизовать число n.

10.107. Даны числа n и m. Вычислите последовательность $a_1, a_2, \ldots, a_n$, где $a_i \equiv i^{-1} \pmod{m}$ за $\mathcal{O}(n + \log m)$.

10.108. Докажите, что $\varphi(n) = \Omega(\sqrt{n})$.

10.109. Докажите, что $\varphi(n) = \Omega\left(\frac{n}{\log n}\right)$.

10.110. Докажите, что $\varphi(n) = \Omega\left(\frac{n}{\log \log n}\right)$.

10.111. Опишите критерий существования решения в китайской теореме об остатках, если модули m_1 и m_2 не являются взаимно-простыми.

Неделя 11. Быстрое преобразование Фурье

Устная часть

11.112. Дано число длины n в a-ичной системе счисления. Переведите его в b-ичную систему счисления за $\mathcal{O}(n \log(\max(a, b)) \log^k n)$ для какого-нибудь k.

11.113. Даны два массива a и b длины n. Найдите скалярное произведение массива a со всеми циклическими сдвигами массива b. Время работы: $\mathcal{O}(n \log n)$.

11.114. Даны две одномерные пластины, разделенные на ячейки длины 1. В некоторых ячейках первой пластины есть выступы, в некоторых ячейках второй — пазы. Найти все сдвиги, при которых можно совместить пластины (для этого все выступы должны войти в пазы). Время работы: $\mathcal{O}(n \log n)$.

11.115. Решите предыдущую задачу для двумерных пластин. Время работы: $\mathcal{O}(nm \log nm)$.

11.116. Дана строка t и шаблон s. Строка t состоит из строчных латинских букв, шаблон s состоит из строчных латинских букв и знаков «?». Найти все наложения s на t, удовлетворяющие шаблону. Время работы: $\mathcal{O}(|\Sigma| \cdot |t| \log|t|)$.

11.117. Решите предыдущую задачу за $\mathcal{O}(|t| \log|t|)$.

11.118. Дано дерево на n вершинах. Для каждого d от 1 до n найдите количество простых путей длины d в данном дереве. Время работы: $\mathcal{O}(n \log^k n)$ для какого-нибудь k.

11.119. Вычислите количество различных AVL-деревьев на n вершинах, имеющих глубину h. Время работы: $\mathcal{O}(h \cdot n^2)$.

11.120. Решите предыдущую задачу за $\mathcal{O}(h \cdot n \log n)$.

11.121. За какое время можно возвести полином степени n в степень k?

11.122. За какое время можно перемножить m многочленов со степенями $n_1, n_2, \ldots, n_m$, такими что $n_1 + n_2 + \ldots + n_m = n$. Приведите оценку, зависящую от n и m.

11.123. Даны n различных чисел $x_0, x_1, \ldots, x_{n-1}$. Постройте многочлен n-й степени, который имеет корни только в заданных точках. Время работы: $\mathcal{O}(n \log^2 n)$.

11.124. Рассмотрим многочлен $C(x) = A(x) + i \cdot B(x)$, где $A(x)$ и $B(x)$ — многочлены с вещественными коэффициентами, а i — мнимая единица. Покажите, как восстановить $FFT(A)$ и $FFT(B)$, зная $FFT(C)$.

11.125. Дан многочлен $A(x)$, где $a_0 \neq 0$. Найдите многочлен $B(x)$, такой что $A(x) \cdot B(x) = 1 + x^n \cdot Q(x)$ для некоторого многочлена $Q(x)$, за $\mathcal{O}(n \log n)$.

11.126. Используя предыдущее задание, вычислите $A(x) \mod B(x)$ за $\mathcal{O}(n \log n)$.