АиСД 2021-22. Четвертый семестр

Задания на практики М3132-М3133

⟨Версия от 1 июня 2024 г.⟩

Темы

1 Паросочетания в двудольных графах **1**
1.1 Практика . . . 2

2 Потоки. Форд-Фалкерсон и Эдмондс-Карп **3**
2.1 Практика . . . 4

3 Потоки. Диниц и все такое **5**
3.1 Практика . . . 6

4 Потоки. Push-Relabel и просто задачи **7**
4.1 Практика . . . 8

5 Потоки минимальной стоимости **9**
5.1 Практика . . . 10

6 Минимальный разрез и задача о назначениях **12**
6.1 Практика . . . 13

7 Геометрия **14**
7.1 Практика . . . 15

8 Extra. Параллельные алгоритмы **16**

9 Теория чисел **17**
9.1 Практика . . . 18

10 Extra. Криптография **19**

11 Extra. Финальное ДЗ **20**

Неделя 1. Паросочетания в двудольных графах

Практика

1.1. Дан граф на n вершинах и m ребрах. За время $\mathcal{O}(n+m)$ найдите паросочетание размера хотя бы $\frac{M}{2}$, где M – размер максимального паросочетания.

1.2. **Повторение (?)**. Разбейте за время $\mathcal{O}(nm)$ все

(a) вершины ориентированного ацикличного графа на минимальное число вершинно-непересекающихся путей

(b) ребра ориентированного ацикличного графа на минимальное число реберно-непересекающихся путей

1.3. **Повторение 2**. Осознайте и докажите, что

(a) размер любого вершинного покрытия не меньше размера любого паросочетания

(b) размер минимального вершинного покрытия равен размеру максимального паросочетания (вспомните алгоритм с лекции)

(c) дополнение любого вершинного покрытия – независимое множество

(d) дополнение любого независимого множества – вершинное покрытие

1.4. **Минимальное реберное покрытие**. Дан граф на n вершинах и m. Выберите за время $\mathcal{O}(nm)$ минимальное число ребер, чтобы каждая вершина была концом хотя бы одного из выбранных ребер.
Подсказка: почитайте про тождества Галлаи.

1.5. За время $\mathcal{O}(n+m)$ проверьте единственность максимального паросочетания в графе, если какое-то максимальное паросочетание уже построено и дано.

1.6. Дан двудольный граф и максимальное паросочетание в нем. За время $\mathcal{O}(n+m)$ проверьте

(a) для каждой вершины, существует ли максимальное паросочетание, покрывающее ее

(b) для каждой вершины, существует ли максимальное паросочетание, не покрывающее ее

(c) для каждого ребра, существует ли максимальное паросочетание, содержащее его

(d) для каждого ребра, существует ли максимальное паросочетание, не содержащее его

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

1.8. Дано поле $n \times n$, некоторые клетки которого удалены. Разместите на этом поле

(a) максимальное количество доминошек 1×2 (доминошки можно поворачивать)

(b) максимальное количество небьющих друг друга коней

(c) максимальное количество небьющих друг друга ладей

Дискреточка

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

1.10. Дефектом множества вершин левой доли A в графе называется $\mathrm{def}(A) = |\mathtt{N}_A| - |A|$, где $\mathtt{N}_A$ – множество соседей вершин из A. Найдите в двудольном графе множество с минимальным дефектом.

1.11. Докажите, что в двудольном графе существует совершенное паросочетание (покрывающее все вершины) тогда и только тогда, когда для любого множества вершин левой доли A верно $|A| \leqslant |N(A)|$.

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

Неделя 2. Потоки. Форд-Фалкерсон и Эдмондс-Карп

Практика

По мотивам лекции

2.1. Докажите, что в алгоритме Эдмондса-Карпа

(a) кратчайшие расстояния до вершин в остаточной сети не уменьшаются
Подсказка: доказательство от противного. Рассмотрите вершину v с кратчайшим расстоянием от s до нее из тех, расстояние до которых уменьшится. Рассмотрите ребро $u \to v$ на этом кратчайшем пути.

(b) время работы равно $\mathcal{O}(nm^2)$

2.2. Докажите, что в алгоритме масштабирования потока время работы равно $\mathcal{O}(m^2 \log C)$, где C – ограничение сверху на пропускные способности ребер.
Подсказка: докажите, что для каждого Δ будет найдено не более m дополняющих путей, рассмотрев некоторый разрез графа.

Простые задачи

Def. *Декомпозицией на пути* потока F называется такое множество потоков F_i, что $F = \sum F_i$ и каждый F_i представляет из себя путь из s в t с одинаковой величиной потока на каждом ребре этого пути.

2.3. Предложите алгоритм декомпозиции потока на пути, работающий за время $\mathcal{O}(\text{poly}(n, m))$.

2.4. Предложите работающий за время $\mathcal{O}(\text{poly}(n, m))$ алгоритм поиска минимального разреза в графе, если вам дан уже построенный на нем максимальный поток.

2.5. Сведите задачу поиска максимального потока в графе из произвольной вершины в произвольную к задаче поиска максимального потока с фиксированным истоком и стоком.

2.6. Докажите, что если в графе G существует поток величины x, то для любого $0 \leqslant y \leqslant x$, существует поток величины y.

2.7. Докажите, что если все пропускные способности целочисленные, то существует максимальный поток, в котором поток по каждому ребру тоже целочисленный.

Остальное

2.8. Решите задачу о максимальном паросочетании в двудольном графе с помощью максимального потока. Как связаны минимальный разрез и минимальное вершинное покрытие?

2.9. Приведите пример сети с вещественными пропускными способностями, в котором алгоритм Форда-Фалкерсона

(a) может никогда не завершиться, но при этом будет получать сколь угодно близкие к верному ответу значения.

(b) может никогда не завершиться и даже не приблизиться к верному ответу на сколь угодно малую величину.

2.10. Определите, единственен ли в графе минимальный разрез, за время $\mathcal{O}(nm^2)$.

2.11. Как изменить алгоритм Форда-Фалкерсона, чтобы он работал для неориентированного графа?

Пояснение: хочется услышать какие-то рассуждения про фиктивные обратные ребра.

2.12. Дан неориентированный граф. Разбейте его

(a) ребра на максимальное число реберно непересекающихся путей из s в t

(b) вершины на максимальное число вершинно непересекающихся путей из s в t

за время $\mathcal{O}(nm^2)$.

2.13. Пусть в сети есть ребра $v \to u$ и $u \to v$. Докажите, что существует максимальный поток, в котором по одному из этих ребер ничего не течет.

2.14. Дано клеточное поле $n \times m$. Некоторые клетки свободны, некоторые заняты горами. Есть два замка. Нужно построить в некоторых клетках стены, так чтобы нельзя было пройти от одного замка к другому (нельзя ходить по горам и стенам). Какое минимальное число клеток надо застроить?

2.15. Дан неориентированный граф. В некоторых вершинах находятся k фабрик и k магазинов. Соедините каждую фабрику с магазином так, чтобы пути не пересекались.

2.16. Есть прямоугольный торт $n \times m$. 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 3. Потоки. Диниц и все такое

Практика

3.1. Постройте граф, на котором алгоритм Диница (его стандартная версия) работает за время $\Omega(n^4)$.

> **Def.** *Циркуляцией* называется поток в сети без выделенного истока (то есть назначение ребрам величины потока $f_{u,v}$ так, чтобы для всех v выполнялось, что входящий поток $\sum\limits_{u\to v} f_{u,v}$ равен исходящему $\sum\limits_{v\to w} f_{v,w}$).

3.2. Дана сеть без выделенного истока и стока. Постройте в данной сети

(a) произвольную циркуляцию

(b) циркуляцию величины хотя бы F (то есть чтобы для каждой вершины выполнялось $\sum\limits_{v\to w} f_{v,w} \geqslant F$

(c) псевдо-циркуляцию с условием $\sum\limits_{v\to w} f_{v,w} - \sum\limits_{u\to v} f_{u,v} = \texttt{balance}_v$, где для каждой вершины v задан свой $\texttt{balance}_v$ (за время $\mathcal{O}(n^2m)$

(d) псевдо-циркуляцию с условием, что поток через каждую вершину v лежит между $\texttt{low}_v$ и $\texttt{high}_v$

3.3. Дана матрица $n \times n$, требуется в каждую клетку поставить

(a) число 0 или 1

(b) число от 0 до k

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 в матрице была максимальна, за время $\mathcal{O}(n^4)$.

3.4. Найдите в неориентированном графе простой путь из вершины u в вершину v, проходящий через вершину w, за время $\mathcal{O}(m^{\frac{3}{2}})$.

3.5. Рассмотрим алгоритм

(a) Форда-Фалкерсона

(b) Эдмондса-Карпа

(c) Диница

запускаемый на графе без обратных ребер. Во сколько раз может ответ, находимый таким алгоритмом, отличаться от настоящего?

3.6. Дана сеть в виде планарного графа (граф на плоскости без пересечений ребер). Требуется найти величину максимального потока из s в t за время $o(nm)$.
Подсказка: посмотрите, как выглядит разрез в таком графе.

3.7. Дана сеть и уже построенный в ней поток (все пропускные способности целочисленные). Пропускную способность одного ребра уменьшили на 1. Предложите алгоритм поиска нового максимального потока, работающий быстрее, чем поиск максимального потока с нуля.

3.8. Дана сеть с целочисленными пропускными способностями. Найдите из всех минимальных разрезов такой, в котором меньше всего ребер.

3.9. Дан граф на n вершинах и m ребрах, а также k грузов, которые требуется перевести из вершины s в вершину t. В каждый момент времени по каждому ребру можно перевозить не более одного груза. За $\mathcal{O}(\mathrm{poly(n, m, k)})$ определите минимальное время, за которое можно перевести все грузы.

3.10. В турнире участвуют n команд, каждая играет с каждой. Некоторые игры уже сыграны, некторые еще нет. За победу команда получает два очка, при ничьей обе команды получают по одному очку. Проверить, может ли первая команда выиграть в турнире?

3.11. Есть ориентированный граф, в каждой вершине число входящих и исходящих ребер одинаково. Пусть в этом графе существует k реберно непересекающихся путей из v в u. Докажите, что в нем также есть k реберно непересекающихся путей из u в v.

3.12. Есть поле $n \times n$, которое нужно полить удобрениями. Это можно делать тремя способами: можно полить одну клетку (i, j), это будет стоить $c_{i,j}$, можно полить целиком горизонталь i, это будет стоить r_i, можно целиком вертикаль j, это будет стоить c_j. Нужно полить все клетки хотя бы по одному разу, минимизировав суммарную стоимость.

Подсказка: сведите задачу к минимальному разрезу.

3.13. Дано n карточек, на i-й карточке записаны числа $l_i, l_i+1, \ldots, r_i$. Нужно собрать набор карточек, на котором были бы записаны все числа от 1 до T по одному разу. Сколько непересекающихся наборов можно собрать?

Неделя 4. Потоки. Push-Relabel и просто задачи

Практика

4.1. **Полное доказательство Push-Relabel**. Докажите, что в алгоритме проталкивания предпотока

(a) для вершины v с $\texttt{excess}(v) > 0$ существует путь в остаточной сети до s

(b) высота вершины не может превысить $2n - 1$

(c) даже если не запрещать такие действия в явном виде, никогда не придется делать $\texttt{relabel}(s)$ и $\texttt{relabel}(t)$

(d) количество насыщающих проталкиваний не превосходит $2nm$

(e) количество ненасыщающих проталкиваний не превосходит $2n^2m + o(n^2m)$

4.2. Приведите пример сети на n вершинах, на которой достигается верхняя граница оценки $h(v) \leqslant 2n - 1$ для какой-то вершины.

4.3. Приведите пример сети на n вершинах и m ребрах, на которой достигается оценка $\Omega(n^2m)$ на время работы алгоритма проталкивания предпотока.

Алгоритм Малхотры-Кумара-Махешвари

> **Def.** Назовем *пропускной способностью вершины* $c(v)$ минимум из $\sum\limits_{v\to u} c_{vu}$ и $\sum\limits_{u\to v} c_{uv}$, то есть максимальную величину потока, который может протекать через вершину v.

4.4. Покажите, что если v_0 – вершина с минимальной пропускной способностью в слоистой сети, то существует поток из s в t величины $c(v_0)$, все пути из декомпозиции которого проходят через v.

4.5. Рассмотрим одну фазу алгоритма Диница. Вместо поиска дополняющего пути с помощью `dfs` будем выбирать вершину с минимальным $c(v)$ и пускать поток из s в t «через v». Сделаем это независимо из v в t и из v в s, идя по ребрам в обратном направлении.
Сформулируйте алгоритм формально, и покажите, что он работает за $\mathcal{O}(n + m_{\text{del}})$, где m_{del} – количество ребер, которые становятся насыщенными.
Подсказка: будем насыщать ребра, исходящие из каждой встреченной вершины, последовательно, пока в ней есть избыток потока.

4.6. В контексте предыдущей задачи покажите, что суммарно одна фаза алгоритма Диница теперь работает за $\mathcal{O}(n^2)$, и весь алгоритм, соответственно, за $\mathcal{O}(n^3)$.

Просто задачи на потоки

4.7. Дана транспортная сеть, в которую не добавили обратные ребра. На ней запустили алгоритм

(a) Форда-Фалкерсона

(b) Эдмондса-Карпа

(c) Диница

(d) Push-Relabel

Насколько найденный максимальный поток может быть меньше реального максимального потока? Приведите оценки снизу и сверху.

4.8. **Реберная теорема Менгера**. Докажите, что минимальное число ребер графа, которые надо удалить, чтобы из s в t не было пути, равно максимальному числу реберно непересекающихся путей из s в t.

4.9. **Вершинная теорема Менгера**. Докажите, что минимальное число вершин графа, которые надо удалить, чтобы из s в t не было пути, равно максимальному числу вершинно непересекающихся путей из s в t (s и t удалять нельзя).

4.10. Дан прямоугольный торт размера $n \times n$, в k клетках находятся вишенки, в k других клетках – клубнички. Разрежьте торт на k частей

(a) произвольного размера за время $\mathcal{O}(n^6)$

(b) произвольного размера за время $\mathcal{O}(n^3)$

(c) равного размера

так, чтобы в каждой части была и вишенка, и клубничка.

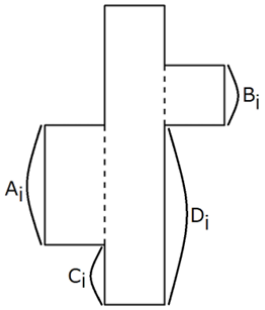
4.11. Дана таблица $n \times m$, в каждой клетке есть растение. Можно полить конкретное растение за $a_{i,j}$ времени, полить всю строку целиком за r_i времени или полить весь столбец целиком за c_j времени. Сколько минимум времени потребуется, чтобы каждое растение было полито хотя бы один раз? Сведите задачу к задаче поиска минимального разреза.

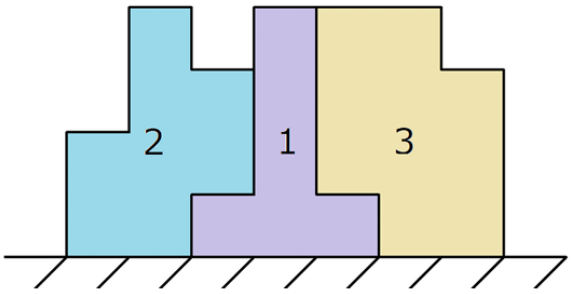
4.12. Пусть сеть представляет из себя планарный граф. Вершины s и t – самая левая и самая правая из вершин, соответственно, а все ребра заданы отрезками на плоскости. Найдите величину максимального потока из s в t за $o(nm)$.

4.13. Дан двудольный граф. За полиномиальное от размера графа время определите минимальное k, при котором можно раскрасить все ребра графа в k цветов, чтобы инцидентные каждой вершине ребра были разных цветов.

4.14. Даны n листов бумаги ширины 3, то есть из трех вертикальных полос:

- ▷ центральная высоты h на расстоянии 0 от низа листа;
- ▷ левая высоты a_i на расстоянии c_i от низа листа ($a_i + c_i \leqslant h$);
- ▷ правая высоты b_i на расстоянии d_i от низа листа ($b_i + d_i \leqslant h$).





Требуется сложить из них фигурку так, чтобы низ каждого листа касался пола, а любая из трех нижних сторон полос каждого листа касалась либо пола, либо другого листа (см. рисунок). Поворачивать листы бумаги нельзя. Определите порядок, в котором их надо расположить, чтобы выполнялись все требования, за время $\mathcal{O}(n \log n)$.

4.15. В некоторой сети уже построен максимальный поток. Покажите, как можно пересчитать максимальный поток при изменении пропускной способности одного ребра на ± 1 быстрее, чем перестроив его целиком с нуля.

Неделя 5. Потоки минимальной стоимости

Практика

5.1. Постройте произвольную сеть без отрицательных циклов

- ▷ на хотя бы 6 вершинах
- ▷ и максимальный поток минимальной стоимости в которой декомпозируется на хотя бы 3 пути.

Проиллюстрируйте на ней работу

(a) классического алгоритма поиска потока минимальной стоимости **Алгоритм**: дополнять вдоль путей минимального веса

(b) алгоритма `cycle cancelling`
Алгоритм: построить произвольный поток максимальной стоимости и дополнять вдоль циклов отрицательного веса

(c) алгоритма `capacity scaling`
Алгоритм: начать с $c_e = 0$ и привести к исходной сети действиями «умножить все c_e на 2» и «увеличить конкретный c_e на 1»

5.2. Постройте пример, аналогичный предыдущей задаче, но содержащий цикл отрицательного веса. Продемонстрируйте работу следующего алгоритма:

- ▷ насытим все ребра отрицательной стоимости, запомним для каждой вершины ее баланс $b(v)$
- ▷ рассмотрим остаточную сеть и создадим новые s' и t'
- ▷ для каждой вершины с $b(v) > 0$ проведем ребро из s' в v пропускной способности $b(v)$ и стомости 0, и для каждой вершины с $b(v) < 0$ проведем аналогичное ребро $v \to t'$
- ▷ найдем максимальный поток минимальной стоимости в полученной сети

Покажите, что

(a) после третьей фазы в сети нет циклов отрицательного веса

(b) при добавлении найденного потока к исходно насыщенным ребрам получается максимальный поток минимальной стоимости в исходной сети

5.3. Докажите, что поток F является потоком минимальной стоимости тогда и только тогда, когда существует способ выбрать потенциалы $\varphi(v)$ таким образом, что

- ▷ если $w^*(e) > 0$, то $f_e = 0$,
- ▷ если $w^*(e) = 0$, то $0 < f_e < c_e$,
- ▷ если $w^*(e) < 0$, то $f_e = c_e$.

5.4. Для трех задач

(a) найти максимальный поток минимальной стоимости

(b) найти циркуляцию минимальной стоимости

(c) найти псевдо-поток с заданным балансом b_v в вершине v для всех вершин в сети с бесконечными пропускными способностями

покажите, что данная задача не проще и не сложнее (то есть что есть сведение в обе стороны) задачи поиска потока величины x минимальной стоимости.

5.5. Алгоритм `cycle cancelling` поиска минимальной циркуляции.

(a) Докажите, что любая циркуляция декомпозируется на не более, чем m циклов, то есть представляется в виде $\sum_{i=1}^{k} C_i$, где C_i – цикл с одинаковой величиной потока на каждом ребре, а $k \leqslant m$.

(b) Пусть уже построена циркуляция C стоимости S, а циркуляция минимальной стоимости имеет стоимость $S_0 < S$. Покажите, что в остаточной сети существует цикл отрицательного веса.

(c) Покажите, что в том же сценарии если дополнить C вдоль отрицательного цикла минимального веса, ее стоимость уменьшится хотя бы на $\frac{S-S_0}{m}$.

(d) Пользуясь этим фактом, покажите, что количество таких шагов до достижения S_0 будет полиномиальным.

(e) Покажите, что если дополнять C вдоль произвольного отрицательного цикла, достижение стоимости S_0 может потребовать более чем полиномиального числа шагов.

5.6. В некоторой сети уже построен максимальный поток минимального веса. Насколько быстро можно его пересчитать

(a) при изменении пропускной способности одного ребра на ± 1?

(b) при изменении стоимости одного ребра на ± 1?

Задачи на mincost потоки

5.7. Задан неориентированный взвешенный граф на n вершинах и m ребрах. Веса ребер неотрицательны. Найдите кратчайший путь из t_1 в t_2, проходящий через s, за время

(a) $\mathcal{O}(\text{poly}(m))$

(b) $\mathcal{O}(m \log n)$

5.8. Дан полный взвешенный двудольный граф, каждая доля которого состоит из n вершин. Найдите в нем совершенное паросочетание минимального веса за время $\mathcal{O}(n^4)$.

5.9. Самолет последовательно посещает города от 1 до n. Для каждой пары (i, j) известно, что $p_{i,j}$ пассажиров хотят сесть на самолет в городе i и сойти в городе j. Каждый из этих $p_{i,j}$ пассажиров готов заплатить $f_{i,j}$ денег за поездку. Найдите максимальную прибыль при условии, что самолет может одновременно вместить не более k человек.
Подсказка: постройте сеть, на которой эта задача сводится к потоку минимальной стоимости.

Задачи на разные потоки

> В этой секции по умолчанию считайте, что каждую задачу требуется решить за полином от размера графа и остальных параметров задачи.

5.10. Дана сеть с целочисленными пропускными способностями. Найдите из всех минимальных разрезов такой, в котором минимальное число ребер.

5.11. Дан неориентированный граф. Требуется ориентировать каждое ребро так, чтобы максимальная исходящая степень была минимальна.

5.12. Дан граф, в вершине s находятся k грузов, которые надо доставить в вершину t. За один ход можно переместить каждый из грузов в соседнюю вершину, но в рамках одного хода по каждому ребру можно перемещать не более одного груза. Найдите минимальное время, за которое можно все грузы доставить в t.

5.13. Есть n человек, n должностей и n хобби. Каждому человеку нравится какое-то свое множество должностей и хобби. Определите, какое максимальное количество непересекающихся троек из человека, должности и хобби можно выбрать.

5.14. Вы играете в настольную игру. В игре есть n типов ресурсов и n производящих их фабрик. За a_i монет можно построить фабрику, производящую i-й ресурс. Также есть m предметов: для каждого известно, какие ресурсы требуются для его производства, а также, что за его продажу можно выручить b_i монет (каждый предмет можно произвести только один раз). Максимизируйте прибыль.

5.15. **Устойчивое паросочетание**. Дан двудольный граф с n вершинами в каждой доле. Для вершины v известен порядок «предпочтений» по ее соседям p_v. Полное паросочетание назовем неустойчивым, если для каких-то двух ребер паросочетания (u_1, v_1) и (u_2, v_2) можно поменять их пары местами (выбрать (u_1, v_2) и (u_2, v_1)), и при этом каждая вершина получит в пару более предпочтительного соседа. Постройте устойчивое паросочетание за время $\mathcal{O}(nm)$.

5.16. Есть n коробок с шариками n различных цветов. В i-й коробке ровно $a_{i,j}$ шариков цвета j. При этом для каждого цвета существует хотя бы один шарик этого цвета. За одну секунду можно переложить один шарик из любой коробки в любую. За какое минимальное время можно сгруппировать шарики в коробках по цветам? Решите за $\mathcal{O}\left(\frac{4}{3}\pi n^3\right)$.

5.17. Дан массив длины n. Требуется выбрать в нем какие-то элементы так, чтобы

- ▷ среди любых m подряд идущих было выбрано не более k;
- ▷ сумма выбранных элементов была максимальна.

5.18. Дана шахматная доска и два расположения нескольких коней на доске. Две фигуры стоять в одной клетке не могут. За минимальное число ходов переведите коней из одного положения в другое.

Неделя 6. Минимальный разрез и задача о назначениях

Практика

Глобальный минимальный разрез

6.1. Дана задача: найти минимальный «разрез» между тремя вершинами a, b и c. То есть удалить минимальное число ребер так, чтобы эти три вершины оказались в трех разных компонентах связности.

(a) Найдем минимальный разрез между a и b, удалим его. После чего найдем минимальный разрез между c и той вершиной из a и b, которая оказалась в его компоненте, и тоже удалим его. Покажите, что такой алгоритм решает исходную задачу неоптимально.

(b) Является ли описанный алгоритм α-приближенным (то есть верно ли для какого-нибудь α, что возвращаемый им ответ всегда не превышает $\alpha \cdot \texttt{opt}$)?

(c) Покажите, что алгоритм остается неоптимальным, если перед его запуском рассматривать все возможные перестановки a, b и c, и после выбирать минимальный из шести найденных ответов.

6.2. Решите задачу 6.1 для планарного графа в случае, когда a, b и c находятся на выпуклой оболочке всех вершин, за время $\mathcal{O}(\text{poly}(n))$.

6.3. Предложите k-оптимальный алгоритм для решения задачи 6.1 для k точек вместо трех, за время $\mathcal{O}(\text{poly}(n, k))$.

6.4. Как изменится время работы и вероятность успеха алгоритма Каргера-Штайна, если

(a) делать три рекурсивных вызова вместо двух?

(b) делать три рекурсивных вызова вместо двух, а первый этап проводить до $\frac{n}{k}$ вместо $\frac{n}{\sqrt{2}}$?

6.5. Дана таблица, в которой записаны числа от 1 до n, каждое ровно по два раза. Определите, какое минимальное число элементов надо переставить, чтобы одинаковые числа стояли в соседних клетках таблицы, за время $\mathcal{O}(\text{poly}(n))$.

6.6. Дана таблица, в каждой клетке которой записано число от 0 до 4. Определите за полиномиальное от размера таблицы время, какие пары соседних клеток соединить, чтобы каждая клетка была соединена с данным количеством соседей.

Венгерский алгоритм

6.7. Дан ориентированный граф со взвешенными ребрами. Покройте все его вершины простыми непересекающимися циклами минимального суммарного веса.

6.8. Дан ориентированный граф. Покройте все его вершины минимальным числом простых непересекающихся циклов за время $\mathcal{O}(3^n)$.

6.9. Дано множество, на элементах которого задан *частичный порядок*. Найдите максимальное по размеру множество попарно несравнимых элементов.

6.10. Сведите задачу поиска минимального по весу взвешенного паросочетания размера n в графе $K_{n,m}$ (при $n < m$) к запуску венгерского алгоритма на графе $K_{m,m}$.

6.11. По заданной квадратной матрице A размера $n \times n$ найдите такие два вектора x и y длины n, что для всех i и j верно $x_i + y_j \geqslant A_{i,j}$ и при этом сумма $\sum\limits_{i=1}^{n} x_i + \sum\limits_{i=1}^{n} y_i$ минимальна.

6.12. Дана квадратная матрица A размера $n \times n$. Требуется увеличить ее элементы так, чтобы для любых i и j выполнялось $A_{i,j} + A_{i+1,j+1} = A_{i,j+1} + A_{i+1,j}$, а суммарное приращение ее элементов было как можно меньше.

6.13. Задан неориентированный граф. Ориентируйте его ребра так, чтобы максимизировать число вершин, у которых входящая степень равна исходящей, за время $\mathcal{O}(n+m)$.

6.14. Задан правильный $(2n + 1)$-угольник. Ориентируйте все его стороны и диагонали, чтобы сумма полученных векторов была равна нулю.

6.15. Определим вес паросочетания как вес максимального ребра в нем. Постройте полное паросочетание минимального веса в двудольном графе.

6.16. Пусть в двудольном графе уже построено максимальное по весу полное паросочетание. После чего вес одного из ребер

(a) уменьшается

(b) увеличивается

Опишите как построить полное паросочетание максимального веса в новом графе быстрее, чем перестраивая его с нуля.

Неделя 7. Геометрия

Практика

7.1. Докажите, что для векторов $v_1 = (\overrightarrow{x_1, y_1})$ и $v_2 = (\overrightarrow{x_2, y_2})$ верно $v_1 \cdot v_2 = x_1x_2 + y_1y_2$.

7.2. Дано n точек на плоскости (x_i, y_i). Постройте за время $\mathcal{O}(n^2)$ «спираль», содержащую все данные точки. Иными словами, упорядочьте точки так, чтобы любые три последовательные точки образовывали левый поворот.

7.3. **Теорема Пика**. Дан многоугольник, каждая вершина которого имеет целочисленные координаты. Обозначим его площадь за S, количество целочисленных точек строго внутри за I, и количество целочисленных точек на границе – за B.

(a) Докажите, что $S = I + \frac{B}{2} - 1$.

(b) Покажите, как найти количество целочисленных точек, лежащих внутри или на границе n-угольника, за время $\mathcal{O}(n)$.

7.4. Дано множество S из n точек с целочисленными координатами (x_i, y_i) на плоскости. Найдите такое множество троек чисел (a_j, b_j, c_j), чтобы

- ▷ для любой точки из S и любой тройки выполнялось $a_jx_i + b_jy_i \leqslant c_j$
- ▷ и для любой целочисленной точки $(x, y) \notin S$ существовала тройка (a_j, b_j, c_j), для которой соответствующее неравенство не выполняется

7.5. Обозначим размер выпуклой оболочки множества из n точек за k. Покажите, как найти выпуклую оболочку за $\mathcal{O}(n \cdot \min(k, \log n))$.

Подсказка: мы рассмотрели два алгоритма, работающие за $\mathcal{O}(nk)$ и $\mathcal{O}(n \log n)$, осталось научиться завершаться за ту же асимптотику, что и наиболее быстрый из них.

Подсказка 2: выпуклая оболочка тут ни при чем, это верно для любых двух алгоритмов, выполняющих одну и ту же задачу.

7.6. **Алгоритм Эндрю**.

(a) Найдите такую точку A, что сортировка точек относительно A по углу равносильна сортировке по y-координате.

(b) Покажите, как можно применить данную идею в алгоритме Грэхэма, чтобы избавиться от необходимости сортировать точки по углу.

7.7. **Алгоритм Чена**. Прочитайте про алгоритм Чена построения выпуклой оболочки. Расскажите его. В каких случаях его использование дает значительный выигрыш по сравнению с алгоритмами Джарвиса и Грэхэма?

7.8. **Convex Hull Trick**. Дана задача динамического программирования с формулой пересчета

$$\texttt{dp}[i] = \max_{k<i}(\texttt{dp}[k] + a_k \cdot x_i + b_k)\,,$$

где $a_{0\ldots n}$, $b_{0\ldots n}$ и $x_{0\ldots n}$ – некоторые параметры из условия.

(a) Сведите поиск данного максимума к нахождению точки на некоторой выпуклой оболочке.

(b) Решите данную задачу динамического программирования за время $\mathcal{O}(n \log n)$, если известно, что $a_i \geqslant a_j$ при $i \geqslant j$.

7.9. Даны n точек на прямой с координатами x_i. Требуется покрыть их m отрезками с минимальной суммой квадратов их длин.

(a) Постройте формулу пересчета динамики для такой задачи.

(b) Сведите ее к формуле, данной в предыдущей задаче, чтобы можно было применить `convex hull trick`.

Неделя 8. Extra. Параллельные алгоритмы

> Задачи с 8.1 по 8.3 стоит воспринимать в модели `Concurrent Read, Exclusive Write PRAM`. За p обозначено число процессоров.
> Почитать про алгоритмы на `PRAM` (в том числе задачи практики) можно здесь.

8.1. Дан массив чисел a размера n. Покажите, как сделать массив, состоящий из всех элементов a без повторений (`std::unique`), за время

(a) $\mathcal{O}\left(\frac{n \log n}{p} + \log^2 n\right)$

(b) $\mathcal{O}\left(\frac{n}{p} + \log n\right)$, если массив отсортирован

8.2. Дан массив из n чисел от 0 до k. Покажите, как построить гистограмму этого массива (то есть посчитать количество вхождений каждого числа) за время

(a) $\mathcal{O}\left(\frac{n \log n}{p} + \log^2 n\right)$
Подсказка: подумайте про сортировку и префиксные суммы.

(b) $\mathcal{O}\left(\frac{nk}{p} + \log n\right)$

(c) $\mathcal{O}\left(\frac{n}{p} + \log n\right)$, если $k = \mathcal{O}(\log n)$
Подсказка: вспомните каскадирование из параллельной вставки в `2-3-tree`.

8.3. Покажите, как реализовать параллельный поиск в ширину, работающий за время $\mathcal{O}\left(\frac{n+m}{p} + \texttt{depth} \cdot \log n\right)$, где `depth` – максимальное расстояние от стартовой вершины до остальных.
Подсказка: стоит обрабатывать вершины по слоям, останется только понять, как по текущему слою построить следующий.

8.4. Есть n процессов, которые выполняют запросы с структуре данных «стек». Стек представляет собой связный список с указателем на вершину. Предложите lock-free реализацию `push()` и `pop()` для такого стека.

8.5. Почему `CAS` более функционален, чем блокировка? Предложите реализацию блокировки через `CAS`.

8.6. **Очередь Майкла-Скотта**. Прочитайте про очередь Майкла-Скотта и раскажите ее нормально (а не как было на лекции).

8.7. **Ближайшие точки на плоскости**. Даны n точек (x_i, y_i). Требуется найти две ближайшие по расстоянию из них.

(a) Покажите, что в прямоугольнике размером $d \times 2d$ может находиться не больше определенной константы точек на расстоянии $\geqslant d$.

(b) Дайте точную оценку на количество точек на расстояни $\geqslant d$ в таком прямоугольнике.

(c-d) Опишите `divide-and-conquer` алгоритм для поиска ближайшей пары точек, работающий за время
(c) $\mathcal{O}(n \log^2 n)$
(d) $\mathcal{O}(n \log n)$

(e) Дайте оценку времени работы данного алгоритма в модели `PRAM`.

8.8. Дайте общую оценку на время работы алгоритма `divide-and-conquer` в `PRAM`, если время его работы в однопроцессорной схеме записывается рекуррентой $T(n) = \texttt{work}(n) + 2T\left(\frac{n}{2}\right)$.

Неделя 9. Теория чисел

▷ Про двоичный алгоритм Евклида можно почитать тут.

Практика

9.1. Посмотрите реализацию двоичного алгоритма Евклида по ссылке выше. Можно ли реализовать расширенный алгоритм Евклида на основе такой реализации?

9.2. Дано n чисел a_i не больше C. Определите,

(a) можно ли выбрать некоторое их подмножество, `gcd` которого будет равно в точности d, за время $\mathcal{O}(n \log C)$

(b) сколько минимум чисел надо удалить, чтобы `gcd` набора увеличился, за время $\mathcal{O}(n \log n + C)$

(c) сколько минимум раз надо увеличить какие-то из чисел на 1, чтобы `gcd` набора стал больше 1, за время $\mathcal{O}(n \frac{C}{\ln C})$

9.3. Найдите

(a) класс решений диофантова уравнения $ax + by = c$ в виде $(x_0 + k \cdot d_x, y_0 + k \cdot d_y)$

(b) количество решений диофантова уравнения, для которых $|x|, |y| \leqslant M$

(c) класс решений уравнения $ax \underset{m}{\equiv} b$

Все пункты должны решаться асимптотически строго быстрее полного перебора.

9.4. Дано натуральное число n. Найдите такие два $x, y \leqslant n$, чтобы $\texttt{lcm}(x, y)$ было максимально, за время $o(n^2)$.

9.5. Покажите, что стандартное решето Эратосфена на самом деле работает за $\mathcal{O}(n \log \log n)$.
Подсказка: стоит воспользоваться тем, что количество простых чисел от 1 до n асимптотически растет как $\frac{n}{\log n}$.

9.6. Дана таблица $d_{i,j}$ размера $n \times m$. Постройте такие массивы a и b, что

(a) $\texttt{gcd}(a_i, b_j) = d_{i,j}$ для всех i, j

(b) $a_i \cdot b_j = d_{i,j}$ для всех i, j

9.7. Даны два числа n и m. Найдите такое a, что $(a \bmod n) \bmod m \neq a \bmod m$ или скажите, что таких не существует.

9.8. Даны три числа a, b и $p \in \mathbb{P}$. Найдите число решений уравнения $x \cdot a^x \equiv b \mod p$ за время $\mathcal{O}(\sqrt{p} \log p)$.

9.9. Прочитайте про **Китайскую теорему об остатках**. Придумайте алгоритм, позволяющий по набору из n пар a_i и p_i найти такое x, что $x \bmod p_i = a_i$ для всех i. Для этого

(a) Покажите, как решить эту задачу при $n = 2$.

(b) Покажите, как по x_0, для которого выполняются первые $n-1$ требований, найти x, что

$$\begin{cases} x \equiv x_0 \mod (p_1 \cdot \ldots \cdot p_{n-1}) \\ x \equiv a_n \mod p_n \end{cases}$$

9.10. Прочитайте про **алгоритм Гарнера**. Сравните его с алгоритмом из предыдущей задачи. Чем алгоритм Гарнера лучше?

9.11. Посчитайте мультиномиальный коэффициент $\binom{n}{a_1,a_2,\ldots,a_m}$ для $\sum\limits_{i=1}^{m} a_i = n$ за время

(a) $\mathcal{O}(n^2)$

(b) $\mathcal{O}(n)$

(∗) $\mathcal{O}(m + \max a_i)$

Вычислить значение надо по модулю некоторого простого числа.

9.12. Вам требуется решить задачу на строки с помощью полиномиального хеширование, однако известно, что коды символов могут быть от 0 до 10^5, а символы, коды которых отличаются на $139k$ для $k \in \mathbb{Z}$, считаются одинаковыми. Как можно строить полиномиальные хеши в таком случае? Обоснуйте ваше решение.

Неделя 10. Extra. Криптография

Прочитайте про основные алгоритмы криптографии, основанные на свойствах теории чисел. В основном понадобится схема RSA.

Также для общего развития можно почитать про схему Эль-Гамаля. Еще мы не рассмотрели алгоритм дискретного логарифмирования: про него можно прочитать здесь.

10.1. Покажите, что в схеме RSA можно восстановить разложение $n = pq$,

(a) зная n и $\varphi(n)$, за время $\mathcal{O}(1)$

(b) зная n, e и d, за время $\mathcal{O}(\log^\alpha n)$

(c) за время $\mathcal{O}(|p - q| \cdot \log^\alpha n)$

10.2. В одной компании для шифрования используется схема RSA, в которой у всех сотрудников n одинаковое, но d и e различные. Сотрудник `A` отправил одно и то же сообщение m двум коллегам. Как злоумышленник может восстановить m, перехватив оба зашифрованных сообщения?

10.3. В другой компании для шифрования используется схема RSA, в которой у всех сотрудников (p, q) различные, но при этом всегда $e = 3$. Сотрудник `A` отправил одно и то же сообщение m трем коллегам. Как злоумышленник может восстановить m, перехватив все три зашифрованных сообщения?

10.4. Прочитайте про защиту от описанных выше атак с помощью добавления случайных битов. Расскажите, в чем суть такой защиты, и как она устроена.

10.5. Два человека `A` и `B` пытаются выбрать секретный ключ k, используя открытый канал. Для этого `A` выбирает случайное число x, а `B` – случайное число y. После чего:

1. `A` выбирает k и отправляет `B` $k_x = k \oplus x$;
2. `B` отправляет `A` $k_{xy} = k_x \oplus y$;
3. `A` отправляет `B` $k_{xyx} = k_{xy} \oplus x$;
4. `B` восстанавливает k как $k_{xyx} \oplus y$.

Почему такая схема ненадежна?

10.6. Три человека хотят выбрать секретный ключ и разбить на три части так, чтобы любые два могли восстановить ключ целиком без участия третьего, но никто из них самостоятельно восстановить ключ не мог бы. Как это сделать?

10.7. Есть n человек и открытый канал связи. Требуется выбрать одного из этих человек «лидером» так, чтобы был достигнут консенсус (все считают лидером одного и того же человека), и при этом чтобы на процесс нельзя было повлиять снаружи. Предложите схему, которая позволяет это реализовать.

Неделя 11. Extra. Финальное ДЗ

★ Правила сдачи

- ▷ Письменное ДЗ оформляется в LaTeXили другом языке разметки (Markdown, Typst); написанные от руки решения приниматься не будут
- ▷ Решения надо прислать **до 11 июня 2024, 23:59** на почту `algorithms.teaching+ct2022@gmail.com`
- ▷ Тема письма должна быть указана в виде «`HWFinal Группа Фамилия Имя`» (без кавычек), например: «`HWFinal M3230 Иванов Иван`»
- ▷ К письму должны быть приложены как сгенерированный `.pdf`-файл, так и исходный `.tex`-файл (или другой исходник в случае другого языка разметки)

11.1. Два человека находятся в точках (x_1, y_1) и (x_2, y_2). Им необходимо встретиться в произвольной точке на прямой $ax + by + c = 0$. Определите минимальное суммарное расстояние, которое им необходимо для этого пройти, за время $\mathcal{O}(1)$.

11.2. Дан выпуклый многоугольник из n вершин, вершины перечислены в порядке следования против часовой стрелки. За время $\mathcal{O}(n)$ найдите *диаметр* многоугольника – расстояние между его двумя наиболее удаленными точками.

11.3. С использованием линейного решета Эратосфена посчитайте для всех натуральных чисел от 1 до n

(a) количество делителей;

(b) функцию Эйлера.

Время работы – $\mathcal{O}(n \log n)$.

(c) Докажите время работы.

11.4. Массив целых чисел a длины n циклически сдвинули влево и поэлементно вычли из исходного. Получился массив с элементами $b_i = a_i - a_{(i+\texttt{shift}) \bmod n}$. По полученному массиву b за время

(a) $\mathcal{O}(n\sqrt{n})$

(b) $\mathcal{O}(n)$

восстановите все возможные значения `shift`, для которых b мог быть получен из некоторого a описанным образом.

11.5. Оцените физическое время работы программы, созданной, чтобы выполнить Birthday attack на хэш MD5.

11.6. Дано полное сбалансированное дерево на n вершинах, у каждой вершины есть вес w_i. Оцените время работы алгоритма поиска максимального по весу паросочетания на этом дереве в модели `CREW PRAM`.