Практика по алгоритмам #13
Графы, MST, DSU

1. Разбор задач из ДЗ #11

2. Разбор задач из ДЗ #12

3. Новые задачи

1. Дана система неравенств на n переменных. Каждое неравенство имеет вид $x_i - x_j \le \delta_{ij}$. Всего неравенств m. Найти решение системы или сказать, что его не существует, за $\mathcal{O}(n \cdot m)$.
2. Пусть все $\delta_{ij} \ge 0$, решить задачу за $o(n \cdot m)$.
3. Выбрать в графе независимое множество размера не менее $\lceil \frac{n}{D+1} \rceil$, где n – количество вершин, D – максимальна степень.
4. А теперь $\lceil \frac{n-1}{D} \rceil$.
5. Проверить, что минимальное по весу остовное дерево единственно. $\mathcal{O}(sort + m)$.
6. Найти второе по весу остовное дерево.
7. Дан орграф, постройте остовное дерево с корнем в вершине 1 минимального веса.
8. Дан взвешанный граф G. Дано минимальное остовное дерево на нем. У ребра e поменяем вес. По графу, остовному дереву, ребру и новому весу найдите новое минимальное остовное дерево за $O(n + m)$.
9. Ребра только добавляются, online, после каждого добавления говорить, является ли граф двудольным
10. В каждой клетке прямой записано число 0 или 1. Поступает информация: четность числа единиц на отрезке $[L_i, R_i]$, найти первый запрос, после которого данные противоречивы.

4. Домашнее задание

4.1. Обязательная часть

1. (1) Пусть мы умеем искать минимум на пути в дереве за время $M(n)$. Дано остовное дерево. За сколько можно проверить то, что оно является минимальным по весу?
2. (3) Дано корневое дерево из n вершин. Все ребра ориентированы к корню. Путь называется вертикальным, если его вершина-конец является предком вершины-начала. На рёбрах дерева есть веса. Даны m вертикальных путей, за $\mathcal{O}((n+m)\log n)$ времени и $\mathcal{O}(n)$ дополнительной памяти с помощью СНМ найти минимум на каждом из путей. **Подсказка:** сперва на 2 балла решите такую же задачу для массива (массив – частный случай дерева!).
3. (3) В неорграф добавляются ребра. Offline. Нужно после каждого запроса добавления говорить, сколько в графе мостов. **Подсказка:** нужно поддерживать дерево компонент реберной двухсвязности.
4. (2) Орграф. Нужно за $\mathcal{O}(n^2)$ обрабатывать запросы "добавить ребор" и "удалить ребро". Гарантируется, что граф всегда остается ацикличным. Также нужно за $\mathcal{O}(1)$ отвечать на запрос "есть ли путь из a в b"? **Подсказка:** наш алгоритм будет вероятностным, давайте поддерживать $c[a, b]$ – количество путей из a в b.
5. (3) Доказать, что если в СНМ использовать только одну эврстику "сжатие путей", то амортизированное время работы будет $\mathcal{O}(\log n)$. **Подсказка:** вспомним про лёгкие и тяжелые ребра. Сколько лёгких ребер на пути? Что происходит с тяжелыми?

4.2. Дополнительная часть

1. (5) Даны n произвольных точек на плоскости, построить MST за $\mathcal{O}(n \log n)$.
2. (3) Построить тест, на котором m запросов к СНМ-у с одной эвристикой "сжатия путей" работают за $\Theta(m \log n)$.