

Алгоритмы и структуры данных. Домашнее задание №8

Выполнил студент Наседкин Дмитрий Сергеевич (группа 242)

Письменная часть

№ 1

Давайте для начала заметим, что если существует правильное разделение на связные области, то можно провести от каждой красной клетки 1 путь до синей и только таких путей тоже будет достаточно, а именно после разбиения на пути можно запустить например **dfs** от клеток очередного пути и походить по свободным еще областям, и так от каждого пути. Очевидно что получится какое-то корректное разделение на области. Доказали эквивалентность этих двух задач(разбиению на непересекающиеся пути и разделение на связные области). (Суммарное время на поиск в глубину $O(nm + k)$).

Будем работать с новой задачей про пути.

Алгос

Теперь сведем нашу задачу к потокам следующим образом:

1. Создадим $2n \times m$ вершин, ij -ой клетке соответствует $v_{ij_{in}}$ и $v_{ij_{out}}$ вершина, и проведем ориентированное ребро из in в out пропускной способности 1 между ними. Это будет означать, что можно зайти в каждую клетку не более одного раза.
2. Каждую из соседних клеток(их 4) для данной, соединим ребрами с пропускной способностью 1, причем из out нашей вершины в in соседней. Это будет означать, что перейти можно только в одну соседнюю клетку + не более одного раза.
3. Создадим исток s и от него проведем в in вершины соответствующие красным клеткам ребра пропускной способности 1.
4. Создадим сток t и от каждой синей вершины out проведем ребро пропускной способности 1 в t .

Построение очевидно работает за $O(nm + k)$.

Теперь на итоговом графе будем искать поток максимальной стоимости. Будем пользоваться Фордом-Фалкерсоном для поиска макс. потока.

Корректность

Очевидно, что максимальный поток здесь будет не больше k , осталось заметить, что если мы нашли макс. поток такого размера(k), то это соответствует правильному разбиению на пути(мы вошли в каждую красную вершину, прошли по каким-то соседним клеточкам, и закончили в синей). В обратную сторону точно также, каждому разбиению на пути можно построить такой поток размера k . То есть построили эквивалентность.

Если макс. поток размера k не найдется, то решения очевидно нет. (Так как если бы оно было то можно построить такой поток, а значит мы бы его нашли).

После алгоритма можно запустить декомпозирование на пути (будет почти тоже самое, что еще один Фалкерсон), путей k , циклов не будет из-за особенностей строения графа, а значит будет работать за время Форда-Фалкерсона.

Форд-Фалкерсон работает за $O(U(V + E))$, где U - размер макс. потока, а (V, E) - кол-во вершин и кол-во ребер в графе соответственно.

Время работы: макс. поток не больше k , а ребер и вершин в нашем графе суммарно $\leq 12nm$ (с запасом), тогда время работы составит $O(nmk)$.

№ 2

(а)

Давайте для каждой вершины левой доли отсортируем список смежности по весам ребер, и возьмем первые a наименьших ребер, заметим, что больше рассматривать и не придется, так как их точно хватит (Вершин слева всего a). Суммарно это работает за время работы сортировки, то есть $O(m \log m)$. Заметим, что теперь в графе $\leq a^2$ ребер.

А теперь запустим `min-cost-max-flow` за $O(U(m + n \log n))$ стандартным сведением поиска макс. парсочка минимального веса к макс. потоку мин. стоимости, где U, n, m - макс. поток, число вершин и ребер соответственно. Очевидно, что у нас макс. поток не больше a .

В нашем случае тогда получим время работы $O(a^3 + a^3 \log a + m \log m)$.