Первый курс, весенний семестр

Практика по алгоритмам #8

Mincost потоки.

Contents

1 Новые задачи

1. **Транспортная задача.**
   В городе есть дороги, заводы-производители и магазины-дистрибьюторы. Дороги образуют орграф, каждая дорога характеризуется своей длиной w_i и максимальной пропускной способностью u_i. i-й завод выпускает A_i единиц товара в день. j-й магазин продаёт B_j единиц товара в день. Нужно составить план доставки товара от заводов к магазинам так, чтобы $\sum_i f_i w_i$ пройденных дорог была минимальна.
2. **mincost + Диниц.**
   Рассмотрим следующий алгоритм для поиска mincost: пока существует дополняющий путь, находим сеть кратчайших путей, ищем максимальный поток в этой сети.
   a) За сколько работает такой алгоритм?
   b) К каким графам его разумно применять?
3. **Mincost matching, остаточная сеть.**
   Рассмотрим полный двудольный граф и паросочетание минимального веса в нем. Построим новый двудольный орграф: в новом графе есть ребро из вершины первой доли i в вершину второй доли j, если $c[i,j] < c[i, pair_i]$. Из $pair_i$ второй доли есть ребро в вершину i первой доли. Могут ли быть циклы в таком графе?
4. **$[L, R]$ через mincost.**
   Найти $[L, R]$-поток, используя mincost flow за время $\mathcal{O}(mincost)$.
5. **Задача 3 из домашнего задания от 18.03.2015**.
   Дан двудольный граф. У вершин есть неотрицательные веса. Вес ребра равен сумме весов его концов. Найдите паросочетание максимального веса.
   a) $\mathcal{O}(V^3)$
   b) $\mathcal{O}(VE)$. Алгоритм Куна.
6. **Подпоследовательности.**
   a) Выбрать k непересекающихся возрастающих подпоследовательностей максимальной суммарной длины. $\mathcal{O}(Polynom(n, k))$.
   b) Дан произвольный ацикличный граф (какой элемент можно брать после какого), у каждой вершины есть вес. Выбрать k вершинно непересекающхся путей так, чтобы сумма весов выбранных вершин была максимальна. $\mathcal{O}(E + kV^2)$.
7. **Задача про автоматы.**
   Есть k автоматов и n заданий. Про каждое задание известны отрезок времени, во сколько его нужно начать делать, во сколько закончить, а также его стоимость. Каждый автомат может выполнять только одно задание в каждый момент времени. Нужно выполнить задания максимальной суммарной стоимости.
8. **Непрерывная цена, обобщённое паросочетание.**
   Дан двудольный граф. Нужно найти обобщённое паросочетание:
   $0 \le f_{ij} \le 1$, $s_i = \sum_i f_{ij} \le a_i$, $t_j = \sum_j f_{ij} \le b_j$. Стоимость обощённого паросочетания равна $\sum_i cost_i s_i^2$ (вместо $\sum_i cost_i s_i$, как было бы в обычном mincost потоке). Минимизировать стоимость максимального по $\sum_i s_i$ паросочетания.

9*. **Кредитные операции - 2.**
   По заданной матрице a_{ij} найти такие вектора x и y, что $x_i + y_j \ge a_{ij}$, а $\sum x_i + \sum y_j \to \min$. Дополнительно известно, что матрица или квадратная, или неотрицательная.

10*. **Быстрый Mincost.**
   Capacity Scaling + внутри по очереди увеличиваем пропусскную способность рёбер. Покажите, что алгоритм работает за $\mathcal{O}(m \log U \cdot S(n, m, U))$

2 Домашнее задание

2.1 Обязательная часть

1. **(2) Mincost [L,R] flow.**
   Предложите алгоритм для поиска LR-потока минимальной стоимости.

2. **(3) Mincost поток нельзя искать bfs-ом.**
   Рассмотрим задачу: найти в невзвешенном орграфе два рёберно непересекающихся пути из s в t минимальной суммарной длины. Мальчик Вася пытается решить задачу так "запустим 2 раза bfs". Покажите, что и если Вася подумал об обратных рёбрах, и если не подумал, его решение некорректно.

3. **(3) Равномерное паросочетание 2.**
   Дана матрица выполнимости – какой рабочий какие работы может выполнять. Распределить работы между рабочими так, чтобы $|V_{opt} - V_{cur}| \to \min$. Здесь $V_{matching}$ – вектор количеств работ, данных рабочим, а $|X - Y|$ – евклидово расстояние между векторами X, Y.
   $V_{opt} = \{\frac{n}{m}, \frac{n}{m}, \dots, \frac{n}{m}\}$, где n – число работа, m – число рабочих.
   a) Свести задачу к потоку минимальной стоимости
   b) Свести задачу к алгоритму Куна

4. **(2) Цикл.**
   Предложите алгоритм за $o(VE)$, который проверяет оптимальность mincost потока.

5. **(3) Потенциалы.**
   Дан взвешенный граф. Возможно, с отрицательными циклами.
   Расставьте вершинам потенциалы так, чтобы минимальный вес ребра был максимально возможным.

6. **(4) Подгон MST.**
   Дан граф G, в нём выделено остовное дерево T. Мы можем уменьшать и увеличивать веса рёбер. Сделать T минимальным по весу остовным деревом. При этом минимизировать суммарное изменение весов рёбер.
   Подсказка: остов минимальный тогда и только тогда, когда вес любого ребра не из остова не меньше максимума на соответствующем пути. Сведите задачу к задаче про $x_i + y_j > a_{ij}$, которая была разобрана на паре.

2.2 Дополнительная часть

1. **Регионы памяти. Расписание выполнения программ.**
   a) **(4)** Есть k регионов памяти и n программ. У каждого есть размер s_i. У каждой программы есть необоходимое ей количество памяти x_j и время выполнения t_j. Каждой программе нужно сопоставить номер региона памяти i_j, в котором она будет выполняться, и отрезок времени выполнения $[l_j, l_j+t_j)$. Для каждого региона памяти отрезки времени выполнения программ не должны пересекаться. Минимизировать $\sum_j l_j$.
   b) **(3)** Усложним задачу. Теперь у каждой программы есть вектор пар $\langle x_j, t_j \rangle$ – если предоставить программе хотя бы x_j памяти, она будет выполняться t_j времени.

2. **(5) Поток в планарном графе.**
   Дана укладка планарного графа. Вершинам сопоставлены точки на плоскости, рёбра – отрезки между вершинами, рёбра не пересекаются. У рёбер есть пропускные способности. Граф неориентированный. Даны две вершины s и t лежащие на одной грани. Задача: за $\mathcal{O}(Dijkstra)$ найти величину максимального потока из s в t.

3. **(6) Крестьяне и поля.**
   Дана матрица $n \times m$. В некоторых клетки горы, в некоторых живут крестьяне, в некоторых поля. Расстояние между клетками (x_1, y_1) и (x_2, y_2) считается без учёта гор: просто $|x_1 - x_2| + |y_1 - y_2|$. Полей не меньше, чем крестьян. Если фиксирован порядок крестьян p, то можно раздать поля следующим алгоритмом: в порядке p каждый крестьянин получает свободное поле, из таких ближайшее, из таких $x \to \min$, из таких $y \to \min$. Задача: выбрать такой порядок p, чтобы сумма расстояний от крестьян до их полей была минимальна.