

SPb HSE, ПАДИИ, 1 курс, осень 2024/25

Практика по алгоритмам #10

dfs. Начало.

21 ноября

Собрано 22 ноября 2024 г. в 15:42

Содержание

1. dfs. Начало.	1
2. Разбор задач практики	3
3. Домашнее задание	5
3.1. Дополнительная часть	5

dfs. Начало.

1. Компактификация

Придумать способ хранения орграфа на $n \leq 2^{16}$ вершинах и m рёбрах так, чтобы он занимал не более $4m+4$ байт памяти, а dfs с ним работал за $\mathcal{O}(m + n \log m)$.

2. Как хранить?

Как хранить граф, чтобы уметь за $\mathcal{O}(1)$ отвечать на запросы четырёх типов: добавить ребро, удалить ребро, связны ли две данные вершины, перебрать всех соседей вершины v за $\mathcal{O}(\deg_v)$.

3. Треугольник

Дан неорграф, посчитать количество треугольников (K_3).

(a) $\mathcal{O}(V^3)$, (b) $\mathcal{O}(VE)$, (c) оптимизации.

4. Цикл через вершину

Дан орграф и вершина v . Найти просто цикл, проходящий через v за $\mathcal{O}(E)$.

5. Дерево

Дан граф. Проверить, является ли деревом. (a) $\mathcal{O}(E)$, (*) (b) $\mathcal{O}(V)$.

6. А вы точно граф?

Даны две операции: $x \rightarrow (2x+1) \bmod N$ и $x \rightarrow (x^2 + 7) \bmod N$. x целое, $N = 10^6 + 3$.

Можно ли из числа 13 получить число 17? Если да, то как?

7. Ямки

В некоторых вершинах орграфа есть ямки.

Нужно дойти из a в b не более двух раз наступив в ямку. Как?

8. Crazy horses

На шахматной доске стоят 2 коня и ещё какие-то фигуры. Нужно, ходя конями по очереди, поменять их местами. В каждой клетке в каждый момент может стоять только одна фигура.

9. В мире перестановок

Даны перестановки p и q из 10 элементов. Можно ли в каком-то порядке применяя к тождественной перестановки p и q , получить перестановку z ?

10. Строкапуты

Дан орграф, на рёбрах написаны буквы. Найдите путь в орграфе, на котором написана ровно данная строка s . За полином.

11. Нечётный цикл

За $\mathcal{O}(V+E)$ найти в неорграфе цикл нечетной длины.

12. Две клики

Есть n человек, между ними есть симметричное отношение дружбы. Разбейте людей на две группы, чтобы в каждой группе все друг с другом попарно дружили.

13. (*) Кластеризация на два кластера.

Даны объекты, и матрица расстояний d_{ij} (непохожести объектов). Нужно разбить объекты на два множества так, чтобы максимальный из диаметров множеств был минимален.

Пример объектов: точки на плоскости.

Пример объектов: тексты и их расстояние Левенштейна.

14. (*) Разбиение на дружелюбные доли

У каждой вершины не более 3 врагов.

Разбить на 2 доли так, чтобы с вершиной в долю попало не более 1 врага. $\mathcal{O}(V+E)$.

Разбор задач практики

1. Компактификация

Храним число рёбер m . Храним рёбра, закодированные в `int32` в отсортированном порядке. Для каждой вершины при переборе рёбер из v делаем один бинпоиск по $v \ll 16$ и с этого места за линию перебираем.

Придумать способ хранения орграфа на $n \leq 2^{16}$ вершинах и m рёбрах так, чтобы он занимал не более $4m+4$ байт памяти, а dfs с ним работал за $\mathcal{O}(m + n \log m)$.

2. Как хранить?

Хранить для каждой вершине `unordered_set` соседей.

3. Треугольник

$\mathcal{O}(V^3)$ — перебираем тройку вершин и матрицей смежности проверяем.

$\mathcal{O}(VE)$ — перебираем вершину и ребро, и матрицей смежности проверяем.

Ещё лучше: $a \rightarrow (b \geq a) \rightarrow (c \geq b)$ (фиксируем a , перебираем два ребра) и матрицей смежности проверяем наличие ac . Чтобы перебирать только рёбра в большие, списки смежности сортированы.

4. Цикл через вершину

`dfs` из вершины v «в себя». Можно модифицировать `dfs`, чтобы он умел в v зайти второй раз. Можно создать v' копию v и `dfs: v \rightarrow v'`.

5. Дерево

Проверить $m = n - 1$, проверить связность. Проверка связности будет работать за $\mathcal{O}(n)$, т.к. мы уже проверили, что рёбер $n - 1$.

6. А вы точно граф?

В условии описан граф из N вершин и $2N$ рёбер. dfs по нему.

7. Ямки

Вершина нового графа $\langle v, k \rangle$ — вершина старого и сколько раз уже были в ямке.

8. Crazy horses

Вершина нового графа $\langle x_1, y_1, x_2, y_2, who \rangle$, где *who* — кто ходит.

9. В мире перестановок

10. Строкапуты

Динамика `dp[v, i]` — можно ли оказаться в вершине v , выписав первые i символов строки s . Начальное состояние $\forall v \text{ dp}[v, 0] = \text{true}$. Конец пути — любая v : `dp[v, |s|] = true`.

11. Нечётный цикл

Красим граф в два цвета. Если видим ребро из текущей, ведущее в вершину такого же цвета — нашли нечетный цикл. Если цикл мы искали dfs-ом, он как раз лежит на стеке рекурсии.

12. Разделение на две клики

Инvertируем все ребра: если между парой вершин нет ребра, добавим, иначе уберем.
 $g[i, j] \hat{=} 1$. Теперь надо разбить на два независимых множества \Leftrightarrow покрасить в два цвета.

Важное замечание: это работает за $\mathcal{O}(V^2)$, в инvertированном графе ребер $V^2 - E$.

13. (*) Кластеризация на два кластера.

Бинпоиск по ответу.

Внутри если $d_{ij} > x$ нужно класть i и j в разные части \Rightarrow раскраска в два цвета.

Более продвинутое решение: сортируем рёбра по возрастанию, добавляем их в таком порядке и СНМ-ом поддерживаем двудольность.

14. (*) Разбиение на дружелюбные доли

Разобьём как-нибудь. Метод локальных оптимизаций: если у какой-то вершины в её доле 2 врага, перекинем её в другую долю.

Каждый раз уменьшается суммарное число рёбер-врагов внутри долей, поэтому надо $\mathcal{O}(E)$ шагов. Чтобы быстро находить плохие вершины, сделаем это dfs-ом.

```
1 void dfs(int v):
2     if (bad(v)):
3         color[v] ^= 1;
4         for (int x : g[v]) dfs(x);
5 for (int v = 0; v < n; v++)
6     dfs(v);
```

Заметим, что тут нет пометок посещенных вершин, но рекурсивные вызовы возникают только после перекидывания вершины, поэтому время $\mathcal{O}(V + E)$.

Домашнее задание

1. (3) Путь через ад

Дан оргграф. В некоторых вершинах живут монстры. Монстры бывают трёх типов. Проходя через некоторые вершины, можно получить иммунитет к некоторым типам монстров (три разных иммунитета). В каждый момент времени каждый из трёх иммунитетов или есть, или нет, копить их нельзя. Иммунитет при встрече с монстром спасает от гибели ровно один раз и пропадает. Можно ли из вершины a дойти до вершины b и не умереть? ($V, E \leq 10^5$).

Подсказка: важно увидеть правильный граф.

2. (2) Покраска в 4 цвета

За $\mathcal{O}^*(2^n)$ проверить, можно ли покрасить граф в 4 цвета.

Обозначение $\mathcal{O}^*(\dots)$ используется, когда полиномиальная часть времени нам не важна.

Подсказка: в два цвета красить очень просто.

3.1. Дополнительная часть

1. (2) Дырявый океан

Поверхность планеты «dfsland» – один большой океан. В этом океане есть острова, в них есть озёра, в озёрах тоже есть острова, в которых тоже озёра... Жители планеты составили карту местности в виде матрицы $w \times h$, где суша обозначена «#», а вода обозначена «.», при этом края матрицы обязательно являются водой. Остров – набор клеток суши, связный по стороне. Острова не касаются по углу. Найдите самую длинную по вложенности цепочку островов-озёр. $w, h \leq 2000$.