Алгоритмы и структуры данных

Графы, DFS, BFS

CS Center, Новосибирск

Графы

$G = \langle V, E \rangle$, $E \subseteq V \times V$

Ориентированные/неоринтированные

Представление графа:

- Матрица смежности – $O(V^2)$ памяти
- Список рёбер – $O(E)$ памяти

Задачи на графы

- Кратчайшие расстояния
  - Задана длина ребер
- Связность

BFS – поиск в ширину

- Начинаем с вершины s
  - Слой $L_0 = \{s\}$, s – посещённая вершина
- Перебираем вершины из слоя L_i
  - Для каждой вершины u перебираем ребра (u, v)
  - v ещё не посещена
    - добавляем её в слой L_{i+1}
    - помечаем её как посещённую

В слое L_i вершины на расстоянии i от s

BFS – поиск в ширину

Можно использовать очередь

- В очереди часть слоя i и часть слоя i+1

- Достаточно уметь для каждой вершины перечислять исходящие из неё рёбра
- Сложность O(|V|+|E|)
- Решает задачи:
  - компоненты связности
  - расстояние в графе с единичными рёбрами

DFS – поиск в глубину

Получается, если в BFS очередь заменить на стек

Цвет вершины c: V → {W, G, B}

Изменение цвета W→G→B

```
DFS(v /* c[v]=W */) {
    c[V] ← G
    for (v, u) E
        if (c[u]==W)
            dfs(u)
    c[V] ← B
}
```

Сложность O(|V|+|E|)

Стек чего?

- Вершина и итератор ребра

Лемма о белом пути:

- Запустив DFS(x) обойдём все вершины y, такие, что есть путь из x в y из белых вершин
- Если нет белого пути в y, то в y не доберётся
- Если есть путь ($x=x_0$, x_1, x_2, … $x_k=y$):
  - предположим не добрались до y
    - после DFS: c[x]=B, c[y]=W
    - после DFS все x_i либо B, либо W
    - тогда есть ребро (x_i, x_{i+1}), c[x_i]=B, c[x_i]=W
      - противоречие

DFS – поиск в глубину

Можно обойтись двумя цветами

Серый цвет нужен для проверки существования циклов

Классификация рёбер:

- tree edges
- forward edges
- backward edges
- cross edges
-