Алгоритмы и модели вычислений.
Задание 12: Алгоритмы на графах I

Сергей Володин, 272 гр.

задано 2014.05.08

Задача 1

Вход: матрица $A\colon N \times N$ смежности неориентированного графа $G = (V, E)$.

1. Алгоритм:

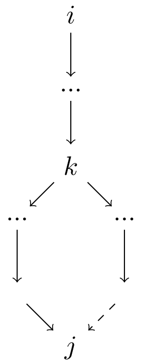
```
int A[NMAX][NMAX];
int color[NMAX];
int visited[NMAX];

#define nextColor(x) ((x + 1) % 2)

int N;

int dfs(int v, int startColor)
{
  if(visited[v])
    return(startColor != color[v]);

  visited[v] = 1;
  color[v] = startColor;

  for(int i = 0; i < N; i++)
    if(i != v && A[i][v])
    {
      if(dfs(i, nextColor(startColor)))
        return(1);
    }

  return(0);
}

int check()
{
  for(int i = 0; i < N; i++)
    visited[i] = 0;

  bool ans = true;

  for(int i = 0; i < N; i++)
    if(!visited[i])
    {
      if(dfs(i, 0)) ans = false;
      break;
    }

  return(ans);
}
```

2. Время работы. В функции `check()` выполняется не более $N = |V|$ поисков в глубину. Каждый выполняется за $O(|V| + |E|)$, поэтому $T(G) = O(|V|^2 + |V||E|) = O(|V|^2 + |V|^3)$. Описание графа $I(G) = c|V|^2$, поэтому $T(I) = O(I^{\frac{3}{2}})$, поэтому алгоритм эффективный.

3. Идея: выполняем обход из каждой непосещенной вершины, для них определяем долю как 0. Если u — в доле m, и ребро $(u, v) \in E$ рассматривается сейчас при обходе, то v в доле $(1 + i) \mod 2$ (в другой). Если возникает противоречие, то не граф двудольный, иначе двудольный, причем найдены доли.

4. Докажем корректность. $P_1(G) = [\text{граф } G \text{ — двудольный}]$, $P_2(G) = [\text{check}()\,|_{G \text{ на входе}} = 1]$. $\neg P_2 \Rightarrow \text{check}() = 0$, так как алгоритм всегда останавливается и выдает 0 либо 1. Поэтому нужно доказать $P_1(G) \Leftrightarrow P_2(G)$

(a) Пусть $P_1(G)$, т.е. граф двудольный. Пусть $\neg P_2(G)$. Тогда один из вызовов $\mathrm{dfs}(i, 0)$ вернул 1. Значит, для некоторой вершины j $\mathrm{visited}[j] = 1$, и $\mathrm{startColor} \neq \mathrm{color}[i]$. Цвет вершины (доля) меняется на противоположный при переходе по ребру, значит, существуют два пути $i \to j$ четной и нечетной длины. Значит, существует простой цикл $j \to k \to j$ нечетной длины. Действительно, пусть $T \subseteq G$ — дерево поиска в глубину (пунктирного ребра в нем нет, но оно из E). Длина цикла $i \to k \to j \to k \to i$ нечетна, если убрать часть $i \to k \to i$ (четной длины), получится простой цикл $j \to k \to j$ нечетной длины. Значит, граф не двудольный (цикл нечетной длины, доли вершин чередуются, получаем ребро между двумя вершинами из одной компоненты связности — противоречие)



Значит, $P_2(G)$ ■

(b) Пусть $P_2(G)$. Найдены множества $V \supseteq L = \{v \in V \big| \mathrm{color}(v) = 0\}$, $R = \{v \in V \big| \mathrm{color}(v) = 1\}$. Пусть $(L^2 \cup R^2) \cap E \neq \varnothing$. Без ограничения общности, $L^2 \cap E \neq \varnothing$. Пусть $(l_1, l_2) \in E$. У l_1 и l_2 одинаковые цвета 0. Без ограничени общности l_2 была найдена первой при поиске. Тогда при поиске в ширину из l_1 был вызов $\mathrm{dfs}(l_2, 1)$, который привел бы к конфликту — противоречие. Значит, это пересечение пусто, и найдены доли графа $\Rightarrow P_1(G)$ ■

Задача 2

Вход: матрица $A\colon n \times n$ смежности неориентированного графа $G = (V, E)$.

1. Идея: выполним обход из каждой вершины, запоминаем посещенные. Те, которые посещены только при текущем обходе — в новой компоненте связности.

2. Алгоритм (каждая компонента связности печатается на отдельной строке)

```
void dfs(int v)
{
  if(visited[v])
    return;

  visited[v] = 1;
  cout << v << " ";

  for(int i = 0; i < N; i++)
    if(i != v && arr[i][v])
      dfs(i);
}

void find()
{
  for(int i = 0; i < N; i++)
    visited[i] = 0;

  for(int i = 0; i < N; i++)
    if(!visited[i])
    {
      dfs(i);
      cout << endl;
    }
}
```

3. Время работы (аналогично задаче 1) $T(I) = O(I^{3/2})$.

4. Корректность. Поиск в ширину находит все вершины, достижимые из v, и только их, т.е. $C(v)$ — класс эквивалентности $C \in V/\sim\colon C \ni v$ (доказано на семинаре). Последующие вызовы dfs производятся только для непосещенных вершин, т.е. для вершин из других компонент связности.

Задача 3

1. Пример:

$s \qquad u \xrightarrow{-1} v$

Из s нет путей, поэтому все $\mathrm{d}[i] = \infty$ — правильный ответ.

2. Пример:

$$\begin{tikzcd} s \arrow[r, bend left, "1"] & u \arrow[l, bend left, "-2"] \end{tikzcd}$$

На первой итерации будет найдено d$[s]$ = 0, d$[u] = \infty$. На второй (непомеченная вершина с минимальным d — s) d$[s] = 0$, d$[u] = 1$, на третьей (непомеченная вершина с минимальным d — u) d$[s] = -1$, d$[u] = 1$, и алгоритм остановится (все вершины помечены). Для s ответ неверный $0 \neq 1$

(каноническое) Задача 51

Идея: модифицируем алгоритм *тот, с тремя циклами* (релаксации). $d(v)$ — максимальная надежность по всем путям $s \to v$.