

Домашнее задание 10-11

Баширов 778

8 мая 2019 г.

1.1

Докажем утверждение сначала в одну сторону: пусть есть некоторое дерево поиска в ширину. Оно будет удовлетворять условиям остоного дерева в задаче.

Теперь в другую: пусть есть некоторое остоное дерево D с зафиксированной точкой s которое удовлетворяет условиям. Докажем от обратного что оно будет деревом некоторого поиска в ширину. Пусть это не так. Поиск будем начинать из точки s . Тогда на некотором шаге поиска будет расхождение деревьев. Пусть на одинаковом расстоянии от s в одном из деревьев присутствует ребро в некую вершину t которого нет в другом дереве. Тогда путь из s в t будет различаться по длине но в обоих деревьях он должен быть кратчайшим.

1.2

Если в подграфе есть пара вершин, не соединенная кратчайшим путем, то ни одна из этих вершин не может быть корнем дерева поиска. Покрыв такими парами все вершины, мы докажем что подграф не является деревом поиска. Найдем такие пары для всех деревьев. Для удобства назовем все вершины слева направо в алфавитном порядке (A,B,C наверху; D,E,F внизу).

1) AD, BF, DC, EF

2) AF, BF, CE, DE

3) AD, BF, CE

Получаем что ни один из подграфов не является деревом поиска.

2

Последовательность действий (порядок соответствует значению):

dA, dD, dG, fG, fD, dE, dF, dH, dI, fI, fH, fF, fE, fA, dB, dC, fC, fB

Ребра деревьев:

AD, DG, AE, EF, FH, HI, BC

Обратные ребра:

CB

Прямые ребра:

EH, FI

Остальные ребра перекрестные

3

1) $\kappa(G) \leq \lambda(G)$

Пусть достаточно удалить k ребер чтобы граф перестал быть связным, тогда удалив по одной вершине (любой) с каждого ребра, мы добьемся такого же результата.

2) $\lambda(G) \leq \delta(G)$

Если удалить все ребра ведущие из некоторой вершины, то граф будет несвязным. Минимальное количество таких ребер – минимальная сте-

пень вершины.

4

Сертификатом будет служить набор вершин(ребер) при удалении которых граф становится несвязным. Несвязность можно проверить поиском в ширину. Построив дерево поиска за полиномиальное время, можно за полиномиальное время сравнить количество вершин в нем с количеством вершин в самом графе.

5

Перебрав все варианты получаем

Точки раздела: *L, I*

Мостов нет

Теперь найдем все двусвязные компоненты

Так как точек раздела две, компонент будет три:

AOBNCDMEL, LKJIF, IGH

6.1

В качестве алгоритма возьмем алгоритм Беллмана-Форда, только подкорректируем функцию релаксации. Возьмем значение вершины с которой мы начинаем поиск равным единице. При релаксации будем умножать значение вершины из которой выходит ребро на значение ребра и присваивать второй вершине. Тогда алгоритм посчитает значение наименее надежного пути для всех вершин. Алгоритм работает эффективно для неплотных графов.

6.3

Наименее надежной сетью будет список(те просто цепочка вершин), так как прохождение сигнала в некоторую вершину возможно одним путем через все вершины с меньшим номером.