Домашнее задание №10-11 (задание пополняется)

Дедлайн: 30 апреля 2019 г., 23:00

Основные задачи

1. (2 + 1 балла)

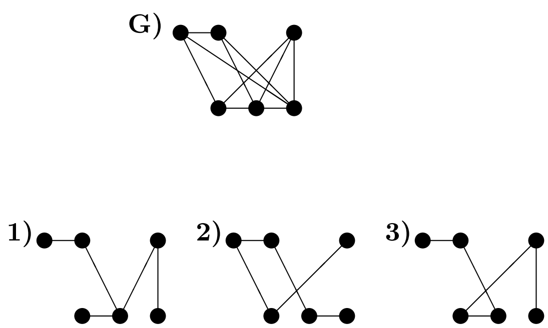
   (i) Докажите или опровергните, что следующее условие дает критерий, когда остовное дерево $F \subseteq G$ является *деревом некоторого поиска в ширину* связного неориентированного графа G.

   Остовное дерево $T \subseteq G$ является деревом некоторого поиска в ширину связного неориентированного графа G, если и только если в нем можно выбрать одну из вершин s за корень так, чтобы T было деревом кратчайших путей из s в графе G. Иными словами, путь по дереву из s в произвольную вершину t содержит не больше ребер, чем кратчайший путь между s и t в G.

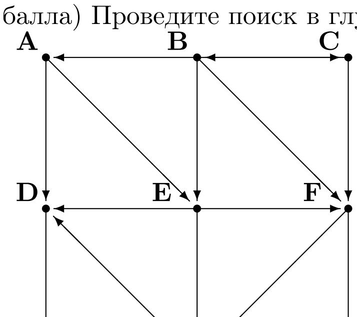
   Если в настоящем виде критерий неверен, то модифицируйте его до корректного.

   (ii) В соответствии с полученным в предыдущем пункте критерием установите, какие из нарисованных деревьев являются деревьями поиска в ширину.

   Формат ответа. Пусть, скажем, критерий верен, тогда при положительном ответе нужно указать корень дерева кратчайших путей, а при отрицательном — для каждого возможного выбора корня нужно указать вершину, расстояние которой до корня в графе меньше, чем соответствующее расстояние по дереву.



2. (2 балла) Проведите поиск в глубину в графе на рисунке.



   Используйте алфавитный порядок вершин. Укажите типы всех дуг графа и вычислите для каждой вершины значение функций $d(\cdot)$ и $f(\cdot)$.

3. ***Связностью* или *вершинной связностью*** $\kappa(G)$ неориентированного графа G называется наименьшее число вершин, удаление которых превращает граф в несвязный или тривиальный. **Реберной связностью** $\lambda(G)$ графаG называется наименьшее число ребер удаление которых превращает граф в несвязный или тривиальный.

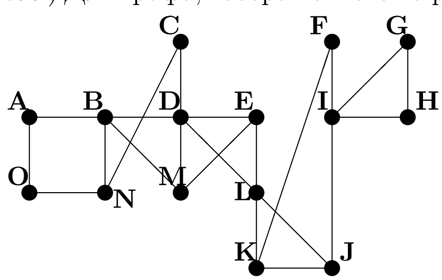
Максимальный по включению k-(реберно) связный подграф графа G называется его k- компонентой (соответственно, k-реберной компонентой). Обычно предполагается, что k-компонента имеет не менее $k+1$ вершин.

(1 балл) Покажите, что для любого G $\kappa(G) \le \lambda(G) \le \delta(G)$ ($\delta(G)$ — это минимальная степень вершин G).

4. ($2+2$ балла) Постройте полиномиальный алгоритм или покажите NP-полноту проверки (i) k-связности и проверки (ii) k-реберной связности графа (k — двоичное число).

5. **Точка раздела** связного неориентированного графа G — это вершина, удаление которой делает граф несвязным. **Мост** — это ребро с аналогичным свойством. **Двусвязная компонента** связного графа содержит ≥ 3 вершин (или ≥ 2 ребер) и состоит из максимального набора ребер, в котором каждая пара ребер принадлежит общему простому (несамопересекающемуся) циклу.

   (1 балл) Для графа, изображенного на рисунке, укажите точки раздела, мосты и двусвязные компоненты.



Дополнительные задачи

k-связность графов

Ниже сформулированы утверждения, которые в принципе можно доказать, используя потоки в сетях, и речь идет только о неориентированных графах.

Вершинной (соответственно, реберной) связностью $\kappa(G)$ (соответственно, реберной $\lambda(G)$) называется наименьшее число вершин (ребер), удаление которых приводит к несвязному или тривиальному графу. .

В предыдущем задании мы установили неравенство $\kappa(G) \le \lambda(G) \le \delta(G)$ ($\delta(G)$ — максимальная степень вершин графа G).

Граф G называется вершинно n-связным или просто n-связным (соответственно, реберноно n-связным), если $\kappa(G) \ge n$ ($\lambda(G) \ge n$). Нетривиальный граф 1-связен, тогда и только тогда, когда он связен, и 2-связен, если и только если в нем более одного ребра и он не имеет точек сочленения. Например, полный граф K_2 не является 2-связным.

Попробуйте в качестве упражнения доказать, что граф двусвязен, тогда и только тогда, когда в нем любые две вершины принадлежят простому циклу.

Теоремы типа Менгера

Пусть u и v — две различные вершины связного графа G. Две простые цепи, соединяющие u и v, называются вершинно-непересекающимися, если у них нет общих вершин, отличных от u и v и реберно-непересекающимися, если у них нет общих ребер. Множество S вершин, ребер или вершин и ребер разделяет u и v, если u и v принадлежят различным различным компонентам графа $G \setminus S$.

Теорема 1 (Карл Менгер (1927))**.** Наименьшее число вершин, разделяющих вершины u и v, равно наибольшеиу числу непересекающихся простых u–v цепей.

Теорема 2 (Форд–Фалкерсон, Элайес–Файнштейн–Шеннон)**.** Для любых двух вершин графа наибольшее число реберно-непересекающихся цепей, соединяющтх их, равно наименьшему числу ребер, разделяющих эти вершины.

Теорема 3 (Хасслер Уитни)**.** Граф n-связен тогда и только тогда, когда любая пара его вершин соединена не менее, чем n вершинно-непесекающимися путями.

Теорема 4. Граф реберно n-связен тогда и только тогда, когда любая пара его вершин соединена не менее, чем n реберно-непесекающимися путями.

Теорема 5. Наибольшее число непересекающихся цепей, соединяющих два непустых непересекающихся вершин V_1 и V_2, равно наименьшему числу вершин, разделяющих V_1 и V_2.

Назовем линией матрицы любую ее строку или столбец. Пусть M — $\{0,1\}$-матрица. Набор единичных элементов матрицы называется независимым, если никакая пара не лежит в общей линии.

Теорема 6. В любой бинарной матрице наибольшее число независимых единичных элементов равно наименьшему числу линий, покрывающих все единицы.

1. $(2+2$ балла) Дана выполнимая 2-КНФ φ, каждый дизъюнкт которой содержит ровно два различных литерала (литерал и его отрицание считаются различными). Будем говорить, что φ 1-*минимальна*, если к ней можно добавить один дизъюнт, содержащий два различных литерала так, чтобы она стала невыполнимой.

   (i) Докажите или опровергните, что следующее условие является критерием 1-минимальности.

   Рассмотрим ориентированный граф G_φ, в котором литералы и их отрицания являются вершинами, а каждый дизъюнкт порождает пару ребер вида: $x \vee y \Rightarrow [e_1 = (\neg x, y), e_2 = (\neg y, x)]$.

   φ является 1-минимальной тогда и только тогда, когда в G_φ есть путь P, соединяющий противоположные литеральные вершины, $x \rightsquigarrow y$, $x = \neg y$ и имеется ребро, ведущее из вершины y в вершину $z \notin P$.

   Если в указанном виде критерий не верен, то дополните его до корректного.

   (ii) Постройте для задачи проверки 1-минимальности как можно более быстрый полиномиальный алгоритм.

   Подсказка. Полезно вспомнить, полиномиальные алгоритмы проверки выпоолнимости 2-КНФ.

2. (3 балла) Постройте линейный по входу алгоритм, который, имея на входе граф G и некоторое его остовное дерево T, определяют, является ли T деревом поиска-в-ширину при старте с некоторой вершины G.

3. $(2 \times 1 + 2 + 1 + 2 \times 2)$ **Линейный алгоритм разбиения графа на двухсвязные компоненты**

   (i) Покажите, что множества вершин, принадлежащие двум разным двусвязным компонентам, либо не пересекаются, либо имеют единственную общую вершину — точку раздела.

   Построим по G новый граф G_b, в котором имеются вершины двух типов: v_a, отвечающие точкам раздела G, и v_b, отвечающие двусвязным компонентам G. Ребра G_b соединяют каждую вершину v_b со всеми вершинами v_a, попадающими в двусвязную компоненту, отвечающую v_b.

   (ii) Покажите, что G_b — дерево, и постройте соответствующее дерево для G из задачи **№ 62**.

   Оказывается, что точки раздела можно находить по дереву поиска в глубину. Затем, опять используя поиск в глубину, можно определить все двусвязные компоненты, т. е. двусвязные компоненты можно находить за линейное время. Мы ограничимся только алгоритмом выделения точек раздела графа.

   (iii) Докажите, что корень дерева поиска в глубину является точкой раздела тогда и только тогда, когда у него больше одного потомка.

   (iv) Постройте контрпример к следующему утверждению из книги **[Кормен 1, задача № 23-2 (б)]**: отличная от корня вершина v дерева поиска в глубину является точкой раздела, если и только если в дереве поиска в глубину не существует обратного ребра от потомка v (включая саму v) до собственного предка v (т. е. отличного от самой v).

   (v) **[Кормен 1, упр. 23-2(в)]**.

   Определим функцию $low(v) = \min[d(v), d(w)$, *если для некоторого потомка и вершины v в G есть обратное ребро* $(u, w)]$.

   Покажите, как вычислить $low(\cdot)$ за время $O(|E|)$ [например, модифицируя поиск в глубину].

   (vi) Покажите, как в линейное время вычислить все двусвязные компоненты графа[1]

[1] **Подсказка.** Сначала, используя решение предыдущих задач, покажите, как с помощью поиска в глубину найти все точки раздела за линейное время. Это и свойства функции $low(\cdot)$ позволяют находить двусвязные компоненты при поиске в глубину, используя дополнительный стек. Можно также находить двусвязные компоненты другим способом (см. **[Кормен I, упр. 23-2(д)–(з)]**).