6 задача: Дополнение до сильносвязного

1 модуль, 2 семестр

ФИВТ МФТИ, 2019

Описание by Илья Белов

1. Текст задачи

Дан ориентированный граф. Определите, какое минимальное число ребер необходимо добавить, чтобы граф стал сильносвязным. В графе возможны петли.

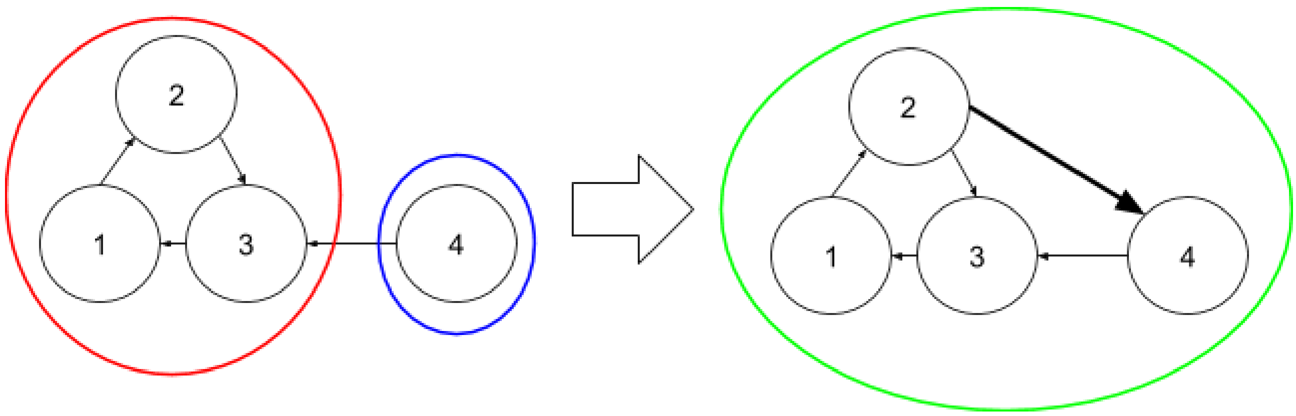
Ввод: в первой строке указывается число вершин графа V, во второй – число ребер E, в последующих – E пар вершин, задающих ребра.

Вывод: Минимальное число ребер k.

Пример:

in	out
4 4 1 2 2 3 3 1 3 4	1

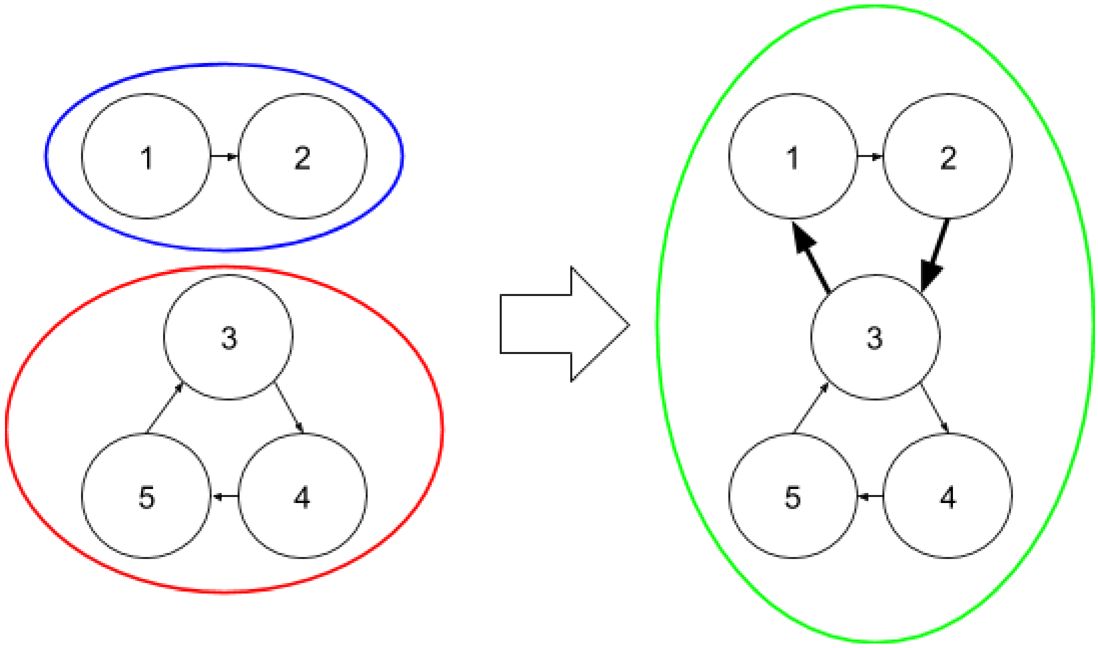
Иллюстрация примера:



Пример:

in	out
5 4 1 2 3 4 4 5 5 3	2

Иллюстрация примера:



2. Описание алгоритма

Конденсируем данный граф с помощью алгоритма Косарайю. У результирующего графа считаем количество стоков (вершины, у которых есть только входящие рёбра) и истоков (вершины, у которых есть только исходящие рёбра). Максимум из этих двух количеств - искомое количество рёбер

3. Доказательство корректности

Доказательство корректности алгоритма Косарайю см. на E-maxx[1]

а) Докажем, что для того, чтобы исходный граф стал сильносвязным, необходимо и достаточно сильной связности

Если исходный граф сильно связен, то его сконденсированный граф состоит из одной вершины. Если некая конденсация графа сильносвязна, её можно сконденсировать далее до одной вершины

б) Теперь докажем корректность выбора количества рёбер.

Необходимость:

Пусть мы выбрали количество стоков n_{sink} (если бы было больше истоков, рассуждения были бы аналогичные). Чтобы граф стал сильносвязным, из каждой вершины должен быть путь в каждую, значит у всех вершин должно быть хотя бы одно исходящее ребро. Следовательно, необходимо добавить хотя бы n_{sink} исходящих рёбер, по одному к каждому стоку

Достаточность:

Покажем алгоритм выбора $max(n_{sink}, n_{source})$ рёбер[2]

Пусть в уже сконденсированном графе $G_{original}$ нет изолированных вершин (случай когда изолированная вершина есть рассмотрим позже). Пусть $T = \{t_1, t_2, ..., t_n\}$ - множество всех стоков и $S = \{s_1, s_2, ..., s_m\}$ - множество всех истоков. Будем считать $n \leq m$, иначе рассуждения аналогичны. Построим двудольный граф $G(T, S)$ по

[1] http://e-maxx.ru/algo/strong_connected_components

[2] Вольный пересказ алгоритма с https://stackoverflow.com/questions/12681785/creating-strongly-connected-components-from-a-dag

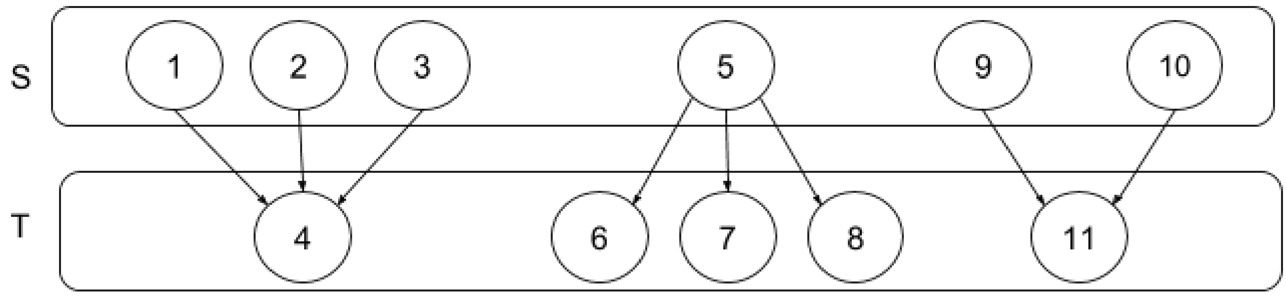
следующему правилу: ребро $(s_j, \ t_i)$ существует только в том случае, если существует путь из s_j и t_i

Пусть M - максимальное паросочетание[3] в $G(T, S)$. Без ограничения общности пусть M состоит из k рёбер: $M = \{(s_1, \ t_1), \ (s_2, \ t_2), \ldots, \ (s_k, \ t_k)\}$. Соединим следующие вершины: $\{(t_1, \ s_2), (t_2, \ s_3), \ldots, (t_{k-1}, \ s_k), (t_k, \ s_1)\}$ и добавим соответствующие рёбра в $G_{original}$. Видно, что каким образом вершины $t_1, ..., \ t_k, \ s_1, ..., s_k$ становятся сильно связными (образуют цикл $t_1 \rightarrow \ s_2 \Rightarrow t_2 \rightarrow ... \rightarrow t_k \rightarrow s_1 \Rightarrow t_1$, где $\Rightarrow$ обозначает цепь из вершин, а не прямое ребро)

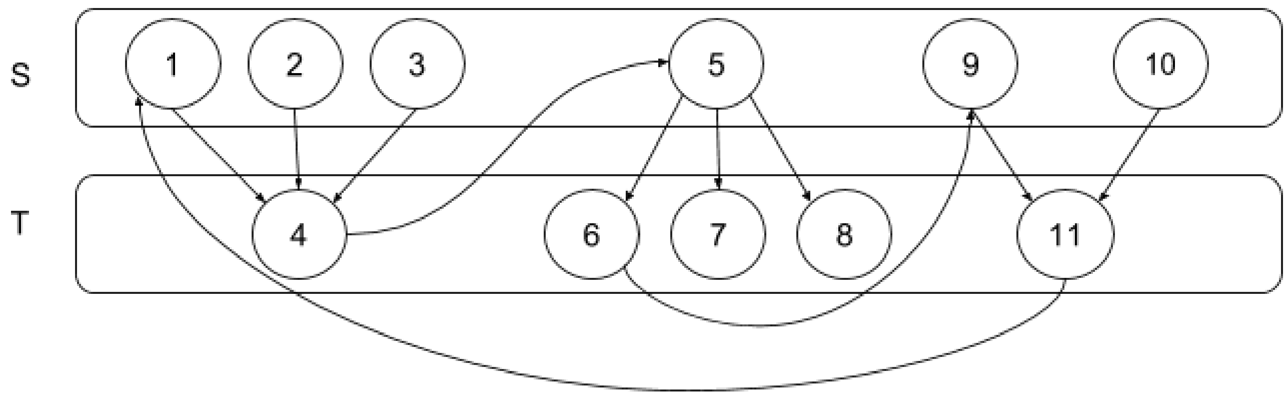
Теперь рассмотрим оставшиеся в $G(T, S)$ вершины. Так как M максимально, каждая вершина в $S - M$ должна быть соединена с вершиной в $T - M$[4]. Так что оставшиеся вершины мы соединяем произвольно, скажем, $\{(t_{k+1}, s_{k+1}), \ldots, (t_n, s_n)\}$ и добавим соответствующие рёбра в $G_{original}$. На каждый оставшийся i-тый исток добавим ребро $(t_j, \ s_i)$, где j - любое. Итоговое добавленное количество рёбер - m

В случае, когда у нас есть изолированная вершина, можно считать её стоком и истоком одновременно, представив эту вершину как две, соединённые ребром

Пример работы алгоритма:



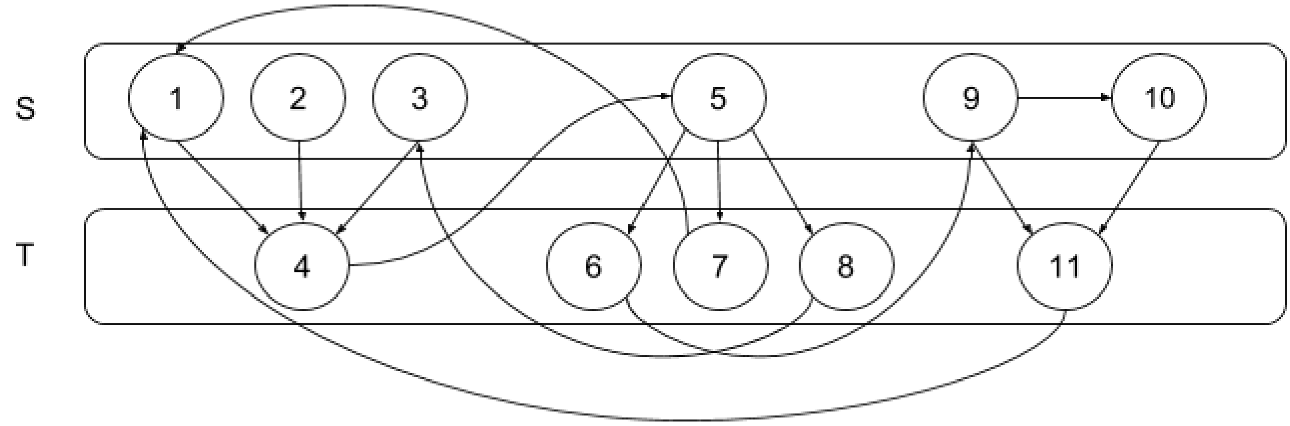
Максимальное паросочетание:
$(1, \ 4), \ (6, 5), \ (9, 11)$
Поэтому добавим рёбра
$(4, \ 5), \ (6, 9), \ (11, 1)$
Получим следующий граф:



Оставшиеся истоки: $\{2, 3, 10\}$, оставшиеся стоки: $\{7, \ 8\}$. Соединим 2 с 7 и 3 с 8: добавим рёбра $(7, \ 1), \ (8, 3)$. Остаётся исток 10, соединим его с любой вершиной, скажем 9. Получим в итоге следующий сильносвязный граф:

---

[3] https://ru.wikipedia.org/wiki/Паросочетание
[4] Я тоже не понял эту часть



Итак, для дополнения графа до сильносвязного необходимо и достаточно $max(n_{sink}, n_{source})$

4. Время работы и дополнительная память

$T = O(V+E)$

$M = O(V)$

5. Доказательство времени работы и дополнительной памяти

Алгоритм представляет собой два вызова DFS, следовательно $T = O(V+E)$. После вызовов DFS происходит ещё два прохода по всем вершинам стоимостью $O(V)$ для поиска стоков и истоков, которые не повлияют на асимптотику. Размер всех контейнеров не превышает $O(V)$, следовательно $M = O(V)$