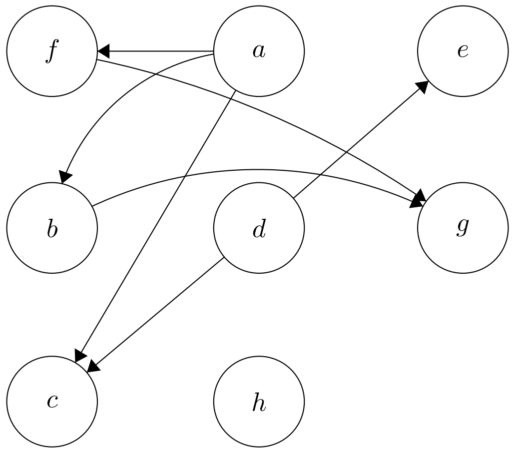
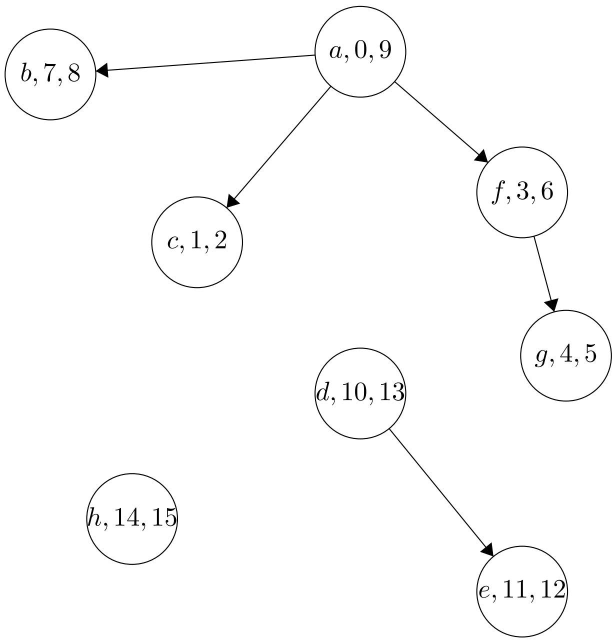
Задание 8. Обходы и кратчайшие пути в графах

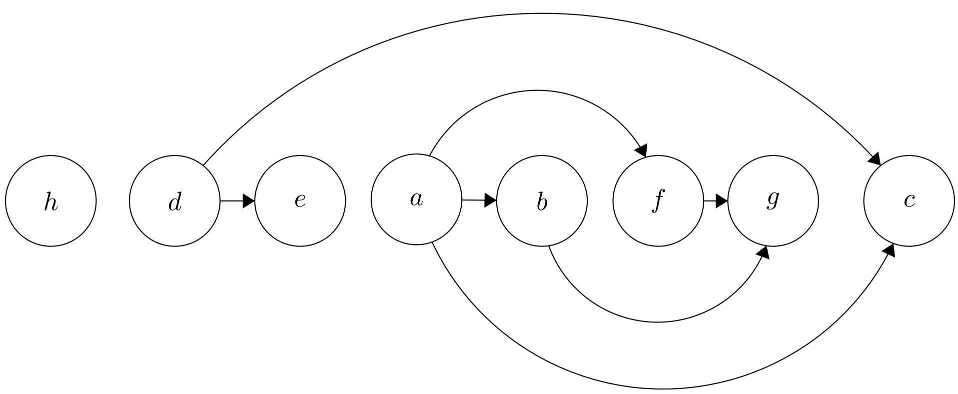
1[2] Сделайте топологическую сортировку графа



$\triangleright$ Лес обхода:

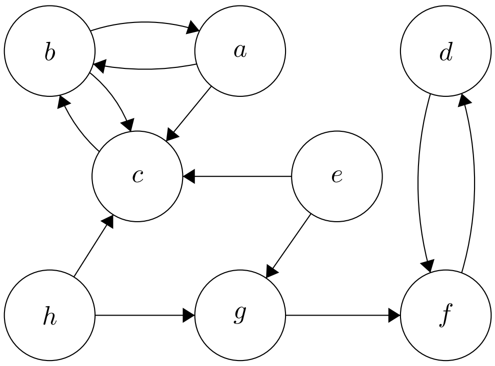


Располагаем вершины по убыванию времени закрытия. Отсортированный граф выглядит следующим образом:

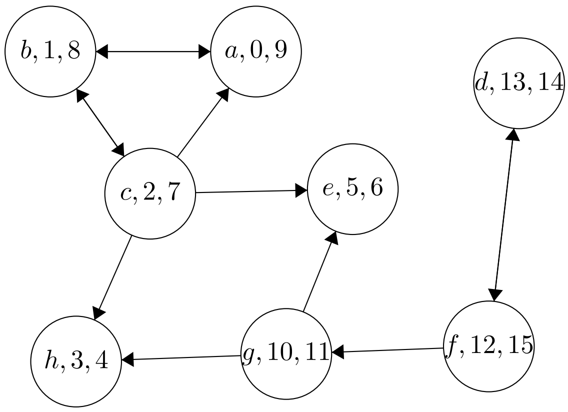


$\triangleleft$

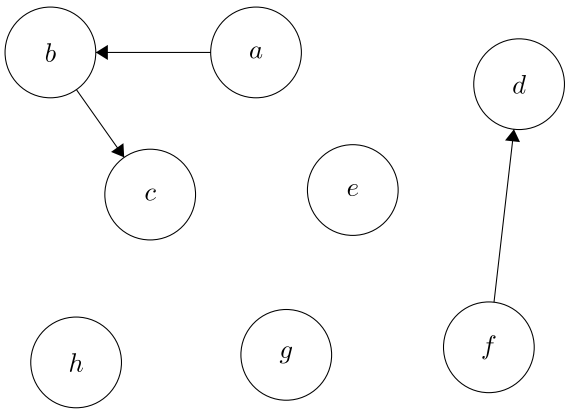
2[2] Найдите конденсат графа



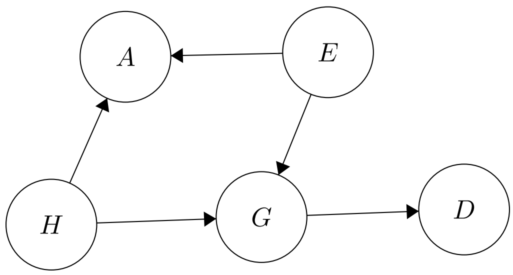
▷ Применим DFS к двойственному графу:



Теперь в исходном графе применим DFS с начальными вершинами, упорядоченными по убыванию времени закрытия. Лес обхода будет состоять из компонент сильной связности.



Конденсат выглядит следующим образом:



◁

3[2 + 2] В графе G был проведён поиск в глубину. Время открытия и закрытия вершин сохранено в массивах d и f. Постройте алгоритм, который на основе этих массивов и описания графа проверяет, является ли ребро e графа G а) прямым ребром б) перекрёстным ребром.

Определения в главе про поиск в глубину в Кормене

$\triangleright$ Рассмотрим лес обхода графа G. Пусть искомое ребро соединяет вершины a и b. Вершина a является предком b тогда и только тогда, когда время открытия вершины b лежит между временем открытия и временем закрытия вершины a. Далее, пусть вершина a имеет k прямых потомков, время открытия t_0, время закрытия t_k. Тогда времена закрытия ее потомков выглядят следующим образом

$$(t_0 + 1, t_1), (t_1 + 1, t_2), \ldots, (t_{k-1} + 1, t_k - 1).$$

a) Итак, сначала необходимо проверить, что время открытия вершины b лежит между временем открытия и закрытия вершины a. Затем, необходимо пройти по вершинам, начиная с времени открытия $t_0 + 1$, так чтобы время открытия следующей было на 1 больше времени открытия предыдущей. Если среди них есть b, то ребро — ребро дерева. Иначе получили **прямое ребро**.

b) Достаточно проверить, вершина a не является ни предком ни потомком вершины b. Это выполняется тогда и только тогда, когда их время открытия каждой вершины не лежит в интервале открытия/закрытия другой. $\triangleleft$

4[1] Рассмотрим алгоритм Дейкстры с модификацией, в которой он сохраняет для каждой вершины предыдущую на кратчайшем пути в массиве предков $\pi[v]$. Докажите, что граф, состоящий из вершин исходного и ребер вида $\pi[v] \to v$, является деревом.

5[1 + 5] В государстве между n городами есть m одностронних дорог. Было решено разделить города государства на наименьшее количество областей так, чтобы внутри каждой области все города были достижимы друг из друга.

1. Предложите эффективный алгоритм, который осуществляет такое разделение, докажите его корректность и оцените асимптотику.

2. Государство решило добиться того, чтобы из каждого города можно было добраться до каждого. В силу бюджетных ограничений было решено построить минимальное число односторонних дорог (любой длины), необходимое для достижения этой цели. Предложите алгоритм, решающий задачу.

6[2] Вам нужно выбраться из лабиринта. Вы не знаете, сколько в нем комнат, и какая у него карта. По всем коридорам можно свободно перемещаться в обе стороны, все комнаты и коридоры выглядят одинаково (комнаты могут отличаться только количеством коридоров). Пусть m - количество коридоров между комнатами. Предложите алгоритм, который находит выход из лабиринта или доказывает, что его нет, за $O(m)$ переходов между комнатами. В вашем расположении имеется неограниченное количество монет, которые вы можете оставлять в комнатах. Минотавр мертв, так что в лабиринте больше никого.

7[3] Дан ориентированный граф на n вершинах ($V = 1, \ldots, n$), который получен из графа-пути (рёбра которого ведут из вершины i в $i + 1$) добавлением ещё каких-то m данных ребер. Найдите количество сильно связных компонент в этом графе за $O(m \log m)$.

8[2] На вход задачи поступает описание двудольного графа $G(L, R, E)$, степень каждой вершины которого равна двум. Необходимо найти максимальное паросочетание в G (которое содержит максимальное количество рёбер). Предложите алгоритм, решающий задачу за $O(|V| + |E|)$.