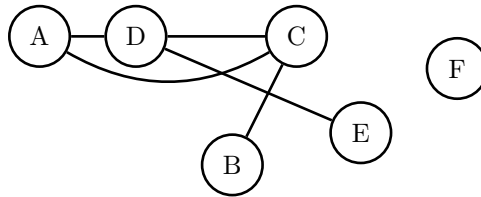
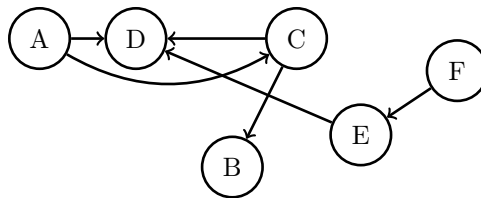


3.2.1. Рассмотрим граф на картинке.



Пусть при его обходе вершины всегда перебираются в алфавитном порядке. В каком порядке будут посещены вершины при обходе поиском в ширину, стартующем в вершине A? Для каждой вершины нас интересует только первое ее посещение.

3.2.2. Рассмотрим ориентированный граф на картинке.



Пусть при его обходе вершины всегда перебираются в алфавитном порядке. В каком порядке будут посещены вершины при обходе поиском в глубину? Для каждой вершины нас интересует только первое ее посещение.

3.2.3. В том же графе в каком порядке будут посещены вершины при обходе поиском в ширину, стартующем в вершине A? Для каждой вершины нас интересует только первое ее посещение.

3.2.4. Докажите, что из всякого связного неориентированного графа можно удалить одну вершину, чтобы он остался связным. Как эффективно находить такую вершину?

3.2.5. Пусть дан неориентированный граф и вершина в нем. Как эффективно проверить, есть ли в графе цикл длины не больше 4, проходящий через эту вершину?

**Замечание.** Если вы решили и сдали все задачи выше во время занятия, вы можете решать и сдавать домашние задачи также во время занятия.

3.2.6. Пусть дан ациклический ориентированный граф. Как эффективно найти топологическую сортировку его вершин?

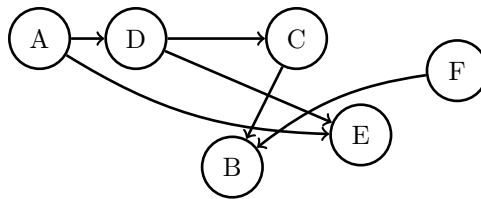
3.2.7. Постройте эффективный алгоритм, который по данному ориентированному графу и ребру в нем проверяет, есть ли в графе цикл, проходящий через это ребро.

**Правила сдачи и оценивания.** Это вторая часть домашнего задания 3. Всего в домашнем задании 6 задач, каждая оценивается в 2,5 баллов. Максимальная оценка за домашнее задание составляет 10 баллов. Если вы наберете больше, то баллы сверх 10 пойдут в виде бонуса в баллы итогового теста.

Дедлайн третьего домашнего задания — 8 июня в 19:00. Решения нужно отправить по адресу [hw.graphs.sber@gmail.com](mailto:hw.graphs.sber@gmail.com). Решения будут проверены до 19:00 9 июня.

Также можно отправить решения до 19:00 6 июня. Тогда они будут проверены до 19:00 7 июня и в случае наличия ошибок можно будет успеть их исправить до основного дедлайна.

**3.2.8.** Рассмотрим ориентированный граф на картинке.



Пусть при его обходе вершины всегда перебираются в алфавитном порядке. В каком порядке будут посещены вершины при обходе поиском в глубину? Для каждой вершины нас интересует только первое ее посещение.

**3.2.9.** В том же графе в каком порядке будут посещены вершины при обходе поиском в ширину, стартующем в вершине A? Для каждой вершины нас интересует только первое ее посещение.

**3.2.10.** Приведите пример ориентированного, сильно связного графа, который перестает быть сильно связным при удалении любой его вершины.