Задача 1.
Решение.

Вариант 1: Нет. Возьмём граф из одного ребра $a \to b$; при обходе в глубину из a в лесе будет одно ребро, а при обходе из b – ноль.

Вариант 2: Да. Лемма 23.10 из Кормена I

Критерии:
I вариант

- неверный контрпример $\leq$**0,25**.
- достаточно простой контрпример без построения лесов (и без их описания): **0,75**

II вариант

- Объяснение того, что любой контур имеет обратное ребро (через ориентацию леса поиска в глубину сверху вних в порядке возрастания времени открытия): **1**
- Близкие, но неполные объяснения без неверных уточнений: **0,75**
- с неверными уточнениями $\leq$ **0,5**
- Объяснение безо всяких доказательств («В цикле обязательно будет обратное ребро»): **0,25**.
- Верное рассуждение без доказательства факта «В каждом контуре есть обратное ребро»: **0,5**.

Задача 2.
Решение.

Вариант 1: Нет. Если все элементы массива различны, то трудоемкость QuickSort в среднем не зависит от числа инверсий во входном массиве и равна $\Theta(n \log n)$.

Вариант 2: Нет. В min-куче минимальный элемент находится в корне, его значение можно узнать за 1 обращение к массиву.

Критерии:
I вариант

- Указано, что трудоемкость не зависит от входящего массива, но не сказано, чему она равна **1**.
- Если же все элементы массива одинаковые, то тогда работает за $\Theta(n^2)$ и ответ – Да. **2**.

II вариант

- Верно расписано для max-кучи **0,5**.

Задача 3.
Решение.

Вариант 1: Нет. Если самое тяжелое ребро графа является мостом, то оно обязано входить в MST.

Вариант 2: Да. Если самое легкое ребро не лежит в каком-то из MST, то при добавлении его в него образуется цикл, в котором можно удалить любое другое ребро и вес дерева уменьшится – противоречие.

Критерии:
II вариант

- Предложен частный случай построения остовного дерева, в который обязательно входит минимальное ребро **1**.
- Сказано без доказательства что в случае различных весов ребер MST единственное + Краскал берет ребро минимального веса **1.5**.

Задача 4.
Решение.

Вариант 1: Да. Доказательство от противного. Рассмотрим минимальное число, на котором перестало выполняться, покажем что в тот момент алгоритм Краскала должен был бы выбрать другое ребро.

Вариант 2: Нет. Контрпример, например, такой: ребра ra, ab с весом 10, rc с весом 11, cd веса 1.

Критерии:
I вариант

- Предъявлено утверждение без доказательства **1**.

- Доказательство неполное **1,5**.

II вариант

- Построен неверный контрпример, неправильно показана работа алгоритма Прима **0,5**.

Задача 5.
В обоих вариантах задача одинакова. Решение.

Нет. С алгоритмом возникают две проблемы: первая связана с определением кратчайшего пути (именно пути, а не его длины). Путь определяется в стандартном алгоритме с помощью ссылки на предка, которая изменяется при релаксации. В случае наличия цикла нулевого веса может произойти некорректная ссылка на предка, приводящая в дальнейшем к невозможности восстановить кратчайший путь из корневой вершины в данную. Наиболее простым примером является петля нулевой длины на произвольной некорневой вершине (назовём её a). Если она окажется последней в списке проверяемых рёбер, произойдёт её релаксация, после этого предком вершины a будет вершина a и восстановить путь от корня до неё невозможно. То же самое может произойти и на нетривиальном цикле нулевой длины (например, треугольнике $a - b - c - a$ с весами $5, -5, 0$).

Вторая проблема возникает на последней (выполняющей проверку на наличие отрицательных циклов) итерации: при наличии нескольких путей равной длины может произойти релаксация ребра. Релаксация ребра на последней итерации в классической реализации алгоритма означает автоматическое наличие цикла отрицательной длины в графе. Алгоритм соответственным образом выдаст ответ “есть цикл отрицательной длины”, что неверно. Такое может произойти, к примеру, на графе $(s \to a)(1), (s \to b)(1), (a \to b)(2)$. Длины путей определятся корректно на второй итерации, но на третьей возможна релаксация ребра $a \to b$ и затем некоерректное завершение алгоритма.

Критерии:

- Предъявлено утверждение о некорректности без доказательства и без примера **1**.
- Доказательство некорректности неполное **1.5**.
- Приведено полное доказательство некорректной работы (любого вида) **2**.
- Предъявлено утверждение о корректности длин кратчайших путей, но не самих путей **0.5**.
- Приведено доказательство корректности длин кратчайших путей, но не самих путей **1**.

Задача 6.
Решение.

Нет. В обоих вариантах первый язык непустой (это тривиально), а второй язык пустой, поэтому сводимости не существует, так как сводящая функция f должна обладать свойством $x \in L_1 \Leftrightarrow f(x) \in L_2$, а $f(x) \in L_2$ неверно.

Доказательство пустоты второго языка:

Вариант 1: Пусть существует плотный несвязный граф на n вершинах, в нём по крайней мере две компоненты, в каждой из них минимальная степень вершин $\geq (n-1)/2$, так что всего вершин не менее $(n-1)/2 + 1 = (n+1)/2$ вершин в каждой компоненте, а всего не менее $n+1$ вершины в графе. Противоречие.

Вариант 2: Пусть существует плотный двудольный граф на n вершинах, в одной из его долей (назовём её A) не более $n/2$ вершин. Минимальная степень вершин другой доли (доли B) не меньше $(n+1)/2$, так что каждая вершина из B должна быть связана по крайней мере с одной вершиной из B (иначе её степень не превосхожила бы мощность A, равную $n/2$), поэтому граф не двудольный. Противоречие.

Критерии:

- Предъявлено доказательство пустоты второго языка без указания наличия/отсутствия сводимости **1**.
- Предъявлено утверждение о том, что полиномиальные языки сводятся друг к другу (с фразой о двух исключениях или без неё), есть проверка полиномиальности языков, на основе этого приведено утверждение о существовании сходимости **0.5**

Задача 7.
Решение.

Вариант 1. Модификация некорректна. Рассмотрим граф на 4 вершинах A, B, C, D с рёбрами AB, BC, AC, BD, CD, A - исток, D - сток. Все пропускные способности равны 1. После первой итерации ФФ если мы могли пройти по пути $ABCD$ и пустить по нему единичный поток. После этого в каждом пути из истока в сток есть насыщенное ребро, однако поток не максимален: можно пустить поток величины 1 по пути ABD и поток величины 1 по пути ACD.

Вариант 2. Заметим, что если по сети идёт максимальный поток, то на любом пути из истока в сток есть насыщенное ребро. Действительно, если все рёбра не насыщены, то поток по этому пути можно увеличить. В графе есть два непересекающихся пути из истока в сток - по низу и по верху, на каждом должно быть насыщенное ребро, то есть насыщенных рёбер хотя бы два. Приведем пример, когда их действительно ровно два. Поставим на верхнем и на нижнем ребре единичные пропускные способности, на остальных рёбрах - двойки. Можно пустить поток 2 - единицу по низу и единицу по верху. Этот поток максимален, так как верхнее и нижнее ребро образуют разрез величины 2 и он насыщает ровно 2 ребра.

Критерии:

Задача 8.

Решение.

Вариант 1. Пусть n – число вершин в G. Если $k < n/2$ Нужно добавить клику на $n - 2k$ вершинах и соединить все вершины этой клики со всеми оставшимися. Получившийся граф содержит клику на $n - k$ вершинах (половине вершин в новом графе) тогда и только тогда, когда исходный граф содержал клику на k вершинах.

Если же $k > n/2$, то нужно добавить $k - n/2$ изолированных вершин.

Вариант 2. Аналогично, добавляем клику на $n - 2k - 1$ вершинах, при $k < n/2$ и $k - n/2$ изолированных вершин, при $k > n/2$.

Обратите внимание, что второй случай существенен: если $k > n/2$, то из существовании клики на $n/2$ вершинах не следует существования клики на k вершинах, поэтому нельзя просто скопировать вход задачи – это не даст сводимость.

Критерии:

- Забыт случай $k > n/2$: **1**.
- Разговоры о том, что нужно добавить вершины без неверых уточнений: **0,25**
- Если сказано как они соединены (анализ первого случая), но корректность не доказана: **0,5**
- Использование клики/независимого множества на k вершинах для построений: **0**
- Перебор всех клик/независимоых множества на k вершинах: **0**