1 Задание 9

1.1 Задача 1

Будем использовать DFS из вершины s. Будем заполнять массив $a[n][2]$ со временем открытия и закрытия, если $a[t][1] = 0$, то вершина не достижима, в другом случае достижима.

Корректность: если вершина достижима, то во время обхода она будет посещена, если $a[t][1] \neq 0$, то она была посещена, а значит достижима.

Асимптотика: время работы совпадает со временем работы алгоритма обхода в глубину, т.е. $O(|V| + |E|)$.

1.2 Задача 2

Будем доказывать по индукции:

База n = 1: в графе нет ребер, а значит есть путь длины $n - 1 = 0$.
Шаг $n = n+1$: пусть нашелся простой путь длины $n-1$, не умаляя общности переименуем вершины, $(u_1, u_2, ..., u_n)$. Будет три случая:

1. Есть ребро (u_n, u_{n+1}), тогда есть простой путь $u_1, u_2, ..., u_n, u_{n+1}$.

2. Есть ребро (u_n, u_1), тошда есть простой путь $u_{n+1}, u_1, u_2, ..., u_n$

3. Нет ранее описанных ребер, т.е. будут ребра $(u_{n+1}, u_n), (u_1, u_{n+1})$. Но тогда найдутся две такие вершины u_k и u_{k+1}, такие что есть ребра (u_k, u_{n+1}) и (u_{n+1}, u_{k+1}). Т.е. будет простой путь $u_1, u_2, ..., u_k, u_{n+1}, u_{k+1}, ..., u_n$.

Выберем первую вершину, будем пользоваться алгоритмом добавления вершин, который был рассмотрен выше.

Корректность: корректность верна из доказательства существования простого пути.

Асимптотика: если реализуются первые два случая, то это будет $O(1)$. Если будет третий, то $O(m)$, где m – количество вершин, находящихся на данный момент в пути, т.е. асимптотика будет $O(|V|^2)$.

1.3 Задача 3

1) Для прямого ребра (u, v) будет верно: $d[v] < d[u]$ и $f[v] > f[u]$ – проверка будет являться ли u потомков v.

Проверим является ли вершина u потомком v, если нет, то ребро не будет прямым, если да, то найдем все смежные с v вершины и проверим, есть ли хоть одна такая вершина, которая является предком u. Если нет, то не является, в другом случае будет являться.

Корректность: существует два типа ребёр, соединяющих предков и потомков: ребра дерева и прямые ребра. Но если проверяемое реберо не ребро дерева, то будет найдена вершина, которая является потомком v и предком u.

Асимптотика: $\theta(deg(v))$ – время работы проверки смежных с v вершин. Т.е. время работы $O(|V|)$.

2) Для перекрестного ребра (v, u) верно: $d[v] > d[u]$ и $f[v] > f[u]$.

Корректность: следует из определения перекрестного ребра: ни одна вершина из v и u не является предком другой. Поэтому та вершина, которая была открыта раньше должна быть и закрыта раньше.

Асимптотика: $O(1)$ – время выполнения двух сравнений.

1.4 Задача 4

Если переформулировать задачу на язык графов, то нам нужно найти компоненты сильной связности. В каждой из них, по определению, любая вершина достижима из другой. А вершина (т.е. город), не лежащей в этой компоненте – не достижим.

Запускаем DFS, транспонируем граф, запускаем DFS по убыванию времени закрытия вершин. Результат каждого запуска DFS после транспонирования – компонента сильной связности.

Корректность: было доказано на лекци..

Асимптотика: два раза запустили DFS, т.е. $O(|V| + |E|)$.

1.5 Задача 5

Используем DFS. Комнаты – это вершины, а коридоры – рёбра. Открывая вершину, будем класть одну монеты, а закрывая вершины будем класть ещё одну монету.

Корректность: алгоритм полностью повторяет dfs, который обойдет все вершины графа.

Асимптотика: Время работы DFS $O(|V|+|E|)$, но нас интересует только количество проходов по коридорам и не интересует время обработки комнаты, то количество проходов по коридорам $O(m)$.

1.6 Задача 6

Создадим массив из n элементов, все элементы в нём будут равны нулю, каждый раз когда добавляется ребро, ведущее от вершины с большим индексом к вершине с меньшим, в ячейку с индексом равным индексу большей вершины уменьшаем на 1, а с меньшим наоборот добавляем 1. Заведем счетчик $t = 0$ и $d = 0$. Идём по массиву и прибавляем значение в ячейке к d. Когда d станет положительной начинаем считать количество ячеек, в которой d положительно. Когда оно таким не окажется добавляем к s значение перестаем считать d и т.д. Тогда ответ будет $n - s + 1$.

Корректность: заметим, что изначально у нас было n сильных компонент связностей. Каждый раз, когда мы добавляем ребро в обратном направлении, то количество компонент сильных связностей уменьшается на разность индексов вершин $+1$, между которыми построили ребро. И мы считаем вершины, которые лежат между концами хоть каких-нибудь добавленных ребер.

Асимптотика: время, которое нужно, чтобы изменить значения в массиве $O(m)$. Все остальные операции будут стоить $O(1)$, т.к. n является константа. Значит суммарное время работы $O(m)$.

1.7 Задача 7

1.8 Задача 8

Воспользуемся алгоритмом поиска эйлерова цикла и добавим в него три дополнительных условия на цвет:

1. Будем красить вершину в цвет ребра, по котором мы в неё пришли.
2. Когда для очередной вершины мы ищем новое ребро для цикла, ребро обязательно должно отличаться цветом от вершины, из которой оно выходит.
3. Самую первую вершину, которубб мы возьмём пометим цветом -1.

Корректность: следует из корректности алгоритма поиска эйлерова цикла, а дополнительные условия гарантируют выполнению дополнительных условий из условия задачи, если степень каждой вершины четная, и мы обязательно найдем подходящее для поиска эйлерова цикла рербо.

Асимптотика: такая же, как и поиска цикла эйлерова цикла $O(|V|+|E|)$