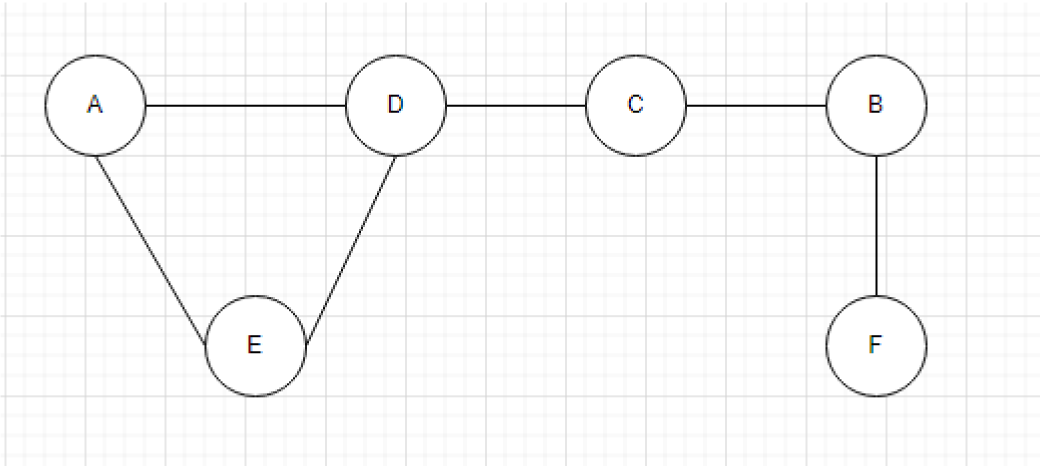
ДЗ №2
Подливальчев И.В.

2.1.8 Дан граф



Пусть при его обходе вершины всегда перебираются в алфавитном порядке.

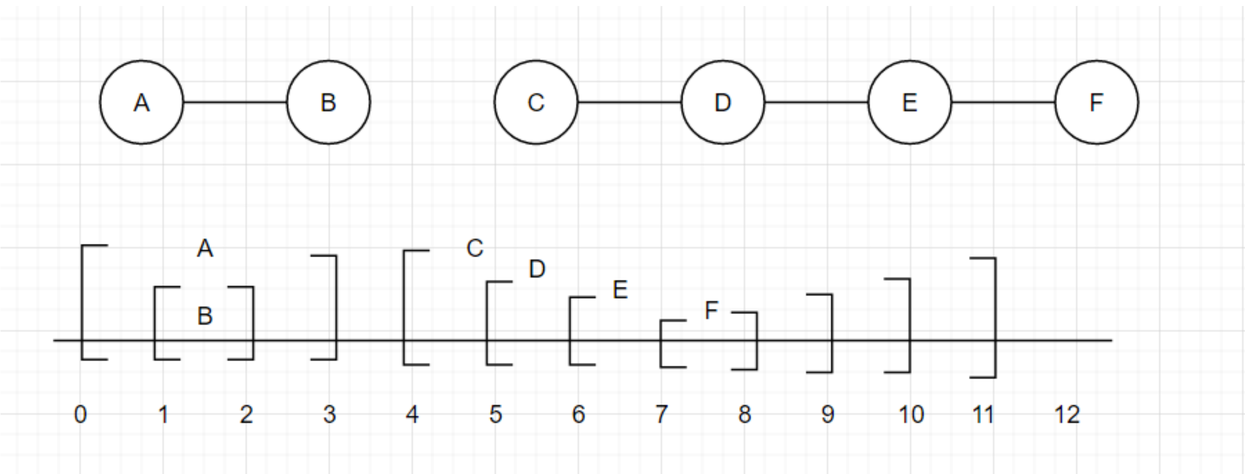
В каком порядке будут посещены вершины алгоритмом DFS. Для каждой вершины интересует только первое её посещение.

Ответ: A – D – C –B –F –E

2.1.9 Пусть для данного графа запускается DSF с нахождением отрезков [pre[v], post[v]] для каждой вершины v.Приведите пример графа на 6 вершинах, в котором оказывается, что для каждого отрезка есть не пересекающийся с ним отрезок. Вершины графа обозначить буквами A, B, C, … и считать, что в поиске в глубину они перебираются в алфавитном порядке. Объясните почему для предложенного графа выполняется условие задачи.

Решение:
В условии задачи не сказано, что в парах (*непересекающийся отрезок*, *отрезок для вершины v*]) *непересекающийся отрезок* не должен повторяться.

Поэтому условию задачи удовлетворяет, например, вот такой граф:

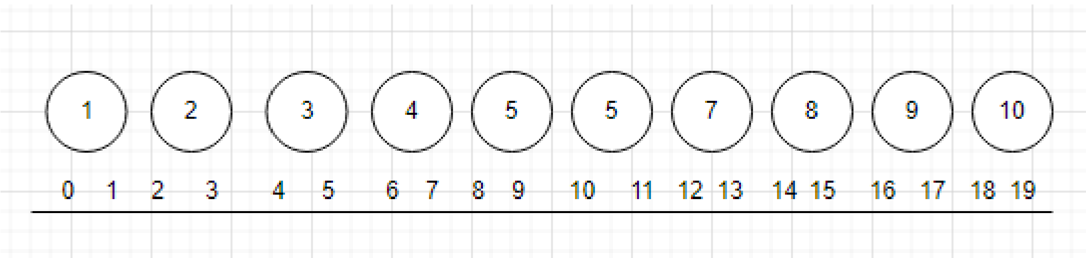


Сопоставление отрезков:
A: C, D, E, F

B: C,D,E,F
C: A,B
D: A,B
E:A,B
F:A,B

В-общем случае, для выполнения условий задачи достаточно иметь граф с двумя непересекающимися отрезками.
[pre[w], post[w]] и *[pre[u], post[u]]*, где:
w принадлежит W,
u принадлежит U
и W и U – непересекающиеся между собой подмножества множества V вершин графа
Тогда для отрезка для любой вершины из множества W можно сопоставить любой из отрезков вершин множества U (напр.отрезок [pre[u], post[u]]), а для отрезка для любой вершины из множества U можно сопоставить любой из отрезков вершин множества W(напр.отрезок [pre[w], post[w]]).
Вышеприведенное условие можно сформулировать короче: Искомый граф должен иметь две области связности, тогда отрезку любой вершины из одной области связности можно сопоставить отрезок любой вершины из другой области связности и отрезки в этих сопоставлениях не будут пересекаться.

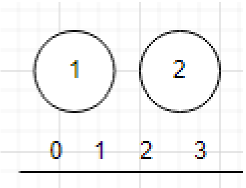
2.1.10. Пусть мы запускаем поиск в глубину в графах на 10 вершинах. Рассмотрим для каждой вершины v отрезок [pre[v], post[v]]
Рассмотрим величину max по v pre[v].
Какое максимальное значение может принимать эта величина? Приведите пример графа, на котором достигается максимальное значение этой величины, и объясните почему оно не может быть ещё больше.

Ответ:
max pre[v] = 2(|V|-1), где |V| - кол-во вершин в графе
для графа на 10 вершинах: max pre[v] = 2(10-1) = **18**
Пример графа:


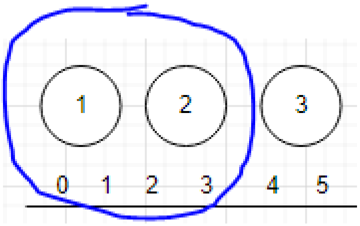
Объясните почему оно не может быть ещё больше : По построению (мы строили граф так, чтобы он максимизировал pre[v]).
Действительно, пусть у нас есть 10 вершин, из них надо построить граф у которого достигалось бы max pre[v].
Начнем строить с первой вершины. Как надо разместить вторую вершину, так чтобы в графе на 2-х вершинах pre[v] было максимальным? Мы можем разместить вторую вершину двумя способами:
а) так чтобы pre[2] было внутри отрезка [pre[1], post[1]], в этом случае pre[1]<pre[2]<post[1]
б) так чтобы pre[2] было вне отрезка [pre[1], post[1]], в этом случае pre[1]<post[1]<pre[2]

Очевидно, что в варианте б) pre[2] > чем pre[2] в варианте а)

Итак, располагаем первые две вершины по варианту б) – см.рисунок ниже:



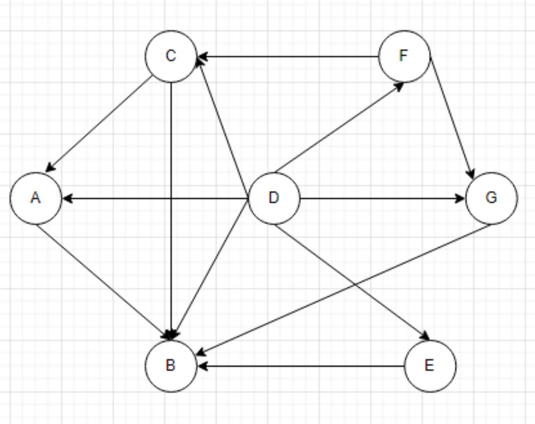
Теперь к получившемуся графу надо добавить третью вершину. Рассуждая аналогичным образом, мы приходим к тому, что для максимизации pre[v] для графа на 3-х вершинах, третью вершину надо расположить вот так (синим выделен граф предыдущего шага):



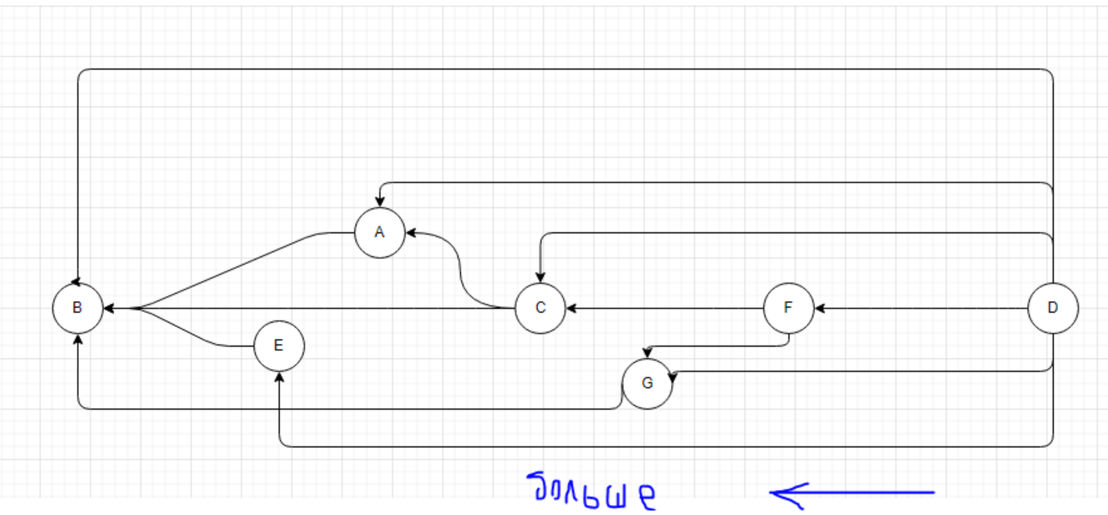
и т.д.

проводя аналогичные рассуждения строим граф на 10 вершинах, максимизирующий pre[v]. Именно этот граф приведен в ответе.

2.2.9 Предъявить топологическую сортировку графа или указать циклЖ



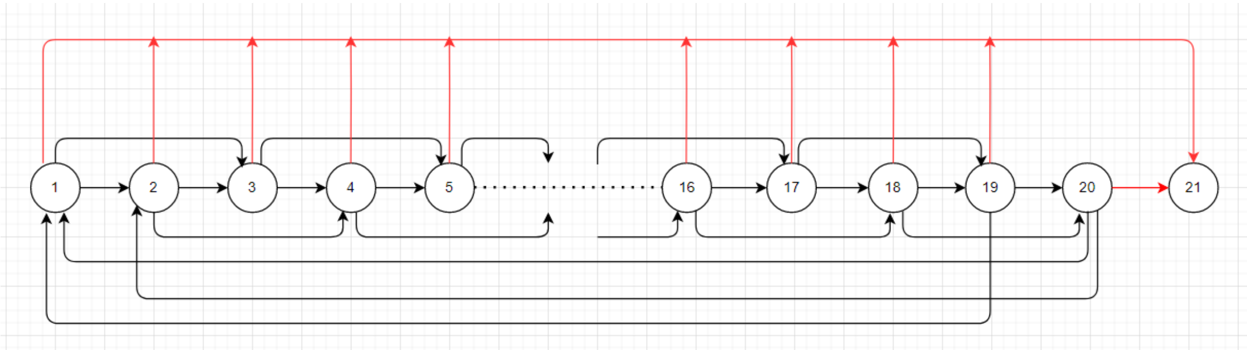
Ответ: от меньшего к большему: DFGCAEB (один из вариантов)



2.2.10. Найти компоненты сильной связности в графе.
Ответ: CBI <- DHA <- E -> FGK

2.2.11. Пусть в ориентированном графе 21 вершина. У каждой из первых 20 вершин входящая степень 2 и исходящая степень 3. Чему равны входящая и исходящая степень 21-й вершины?

Решение:



Ответ: Входящая степень = 20, Исходящая степень = 0