



Дивизион D. День 4 (Разбор)
Графы (BFS и Алгоритм Дейкстры)

Лектор: Дмитрий Руденко

Задача А

- 1) Запускаем BFS из стартовой точки
- 2) Считаем что ячейки с 1, уже used до начала BFS

Задача А

- 1) Запускаем BFS из стартовой точки
- 2) Считаем что ячейки с 1, уже used до начала BFS

Задача А

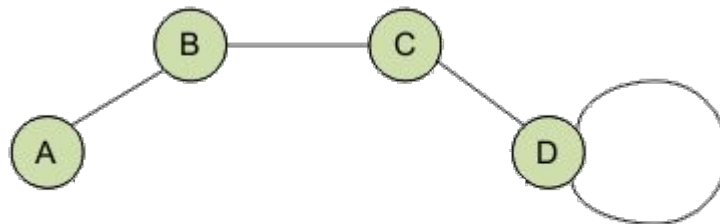
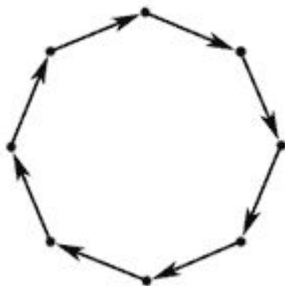
- 1) Запускаем BFS из стартовой точки
- 2) Считаем что ячейки с 1, уже used до начала BFS

Задача В

- 1) 0-1 BFS с восстановлением ответа
- 2) Если вершины принадлежат одному и тому же барону то вес ребра между ними 0, иначе - 1

Задача С

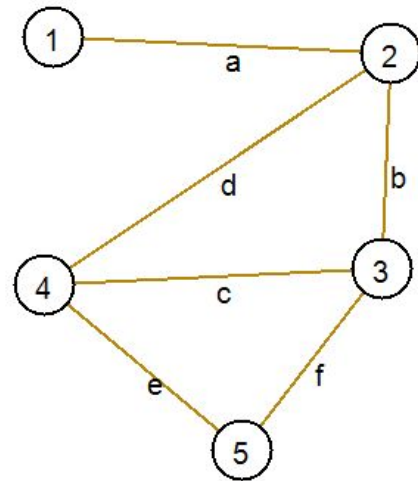
- 1) Между вершинами А и В есть ребро (ориентированное), только в том случае если ключ от В лежит в А
- 2) Ответом будет количество простых циклов + количество вершин в которые входит 0 ребер



Задача Е

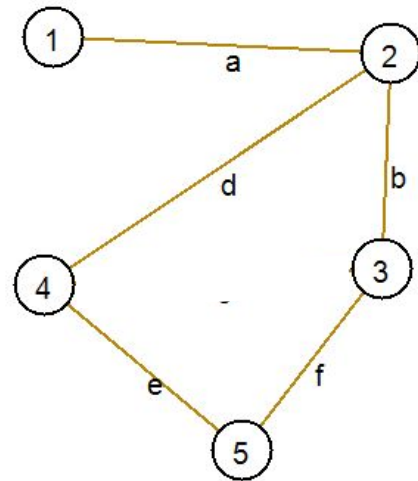
Задача E

- 1) Давайте перебирать ребра которые соединяют вершины a и b
- 2) Удаляем ребро
- 3) Находим кратчайшее расстояние от a до b
- 4) Если оно есть, то длина кратчайшего цикла равна этому расстоянию + вес ребра a->b



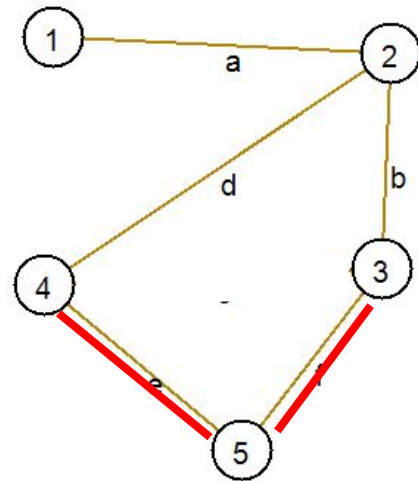
Задача E

- 1) Давайте перебирать ребра которые соединяют вершины a и b
- 2) Удаляем ребро
- 3) Находим кратчайшее расстояние от a до b
- 4) Если оно есть, то длина кратчайшего цикла равна этому расстоянию + вес ребра a->b



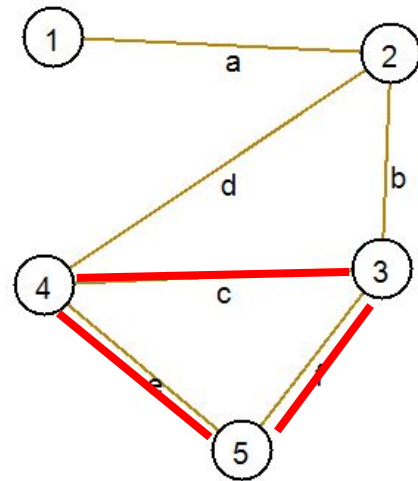
Задача E

- 1) Давайте перебирать ребра которые соединяют вершины a и b
- 2) Удаляем ребро
- 3) Находим кратчайшее расстояние от a до b
- 4) Если оно есть, то длина кратчайшего цикла равна этому расстоянию + вес ребра a->b



Задача D

- 1) Давайте перебирать ребра которые соединяют вершины a и b
- 2) Удаляем ребро
- 3) Находим кратчайшее расстояние от a до b
- 4) Если оно есть, то длина кратчайшего цикла равна этому расстоянию + вес ребра a->b



Задача F

1) Написать Дейкстру :)

Задача G

- 1) Написать Дейкстру с восстановлением ответа :)

Задача Н

- 1) Пусть стоимость бензина в вершине A равняется w
- 2) Тогда все ребра из A в смежные вершины имеют вес w
- 3) Пишем дейкстру :)

Задача I

```
struct Stop {
    int from, start_time, to, finish_time;
};

...

dist[from - 1] = 0;

sort(stops.begin(), stops.end(), [](Stop& left, Stop& right) {
    return left.start_time < right.start_time;
});

for (int i = 0; i < m; i++) {
    if (stops[i].start_time >= dist[stops[i].from - 1]) {
        dist[stops[i].to - 1] = min(dist[stops[i].to - 1], stops[i].finish_time);
    }
}
```

Задача H

- 1) Давайте научимся находить время перевозки C чашек
- 2) Можно найти дейкстрой, просто запретим переходы по вершинам `saracity` которых меньше C
- 3) Найдем ответ бинарным поиском
- 4) Доказательство: с увеличением кол-ва чашек, время только увеличивается, поскольку блокируются пути и кратчайший путь может только ухудшиться
- 5) Имеем монотонную функцию - `bin search challenge accepted`
- 6) $\log(10^7) * 250000$