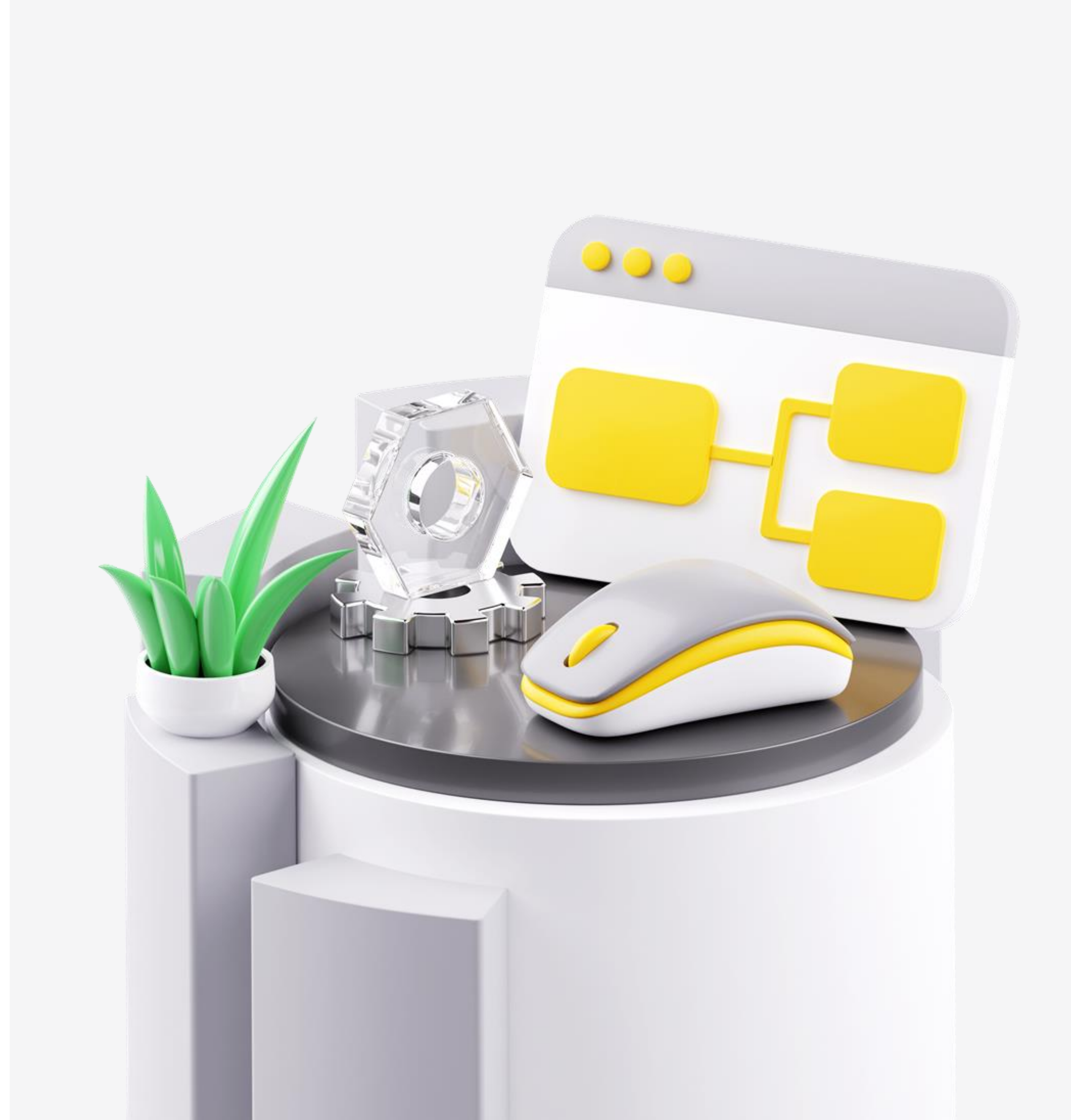
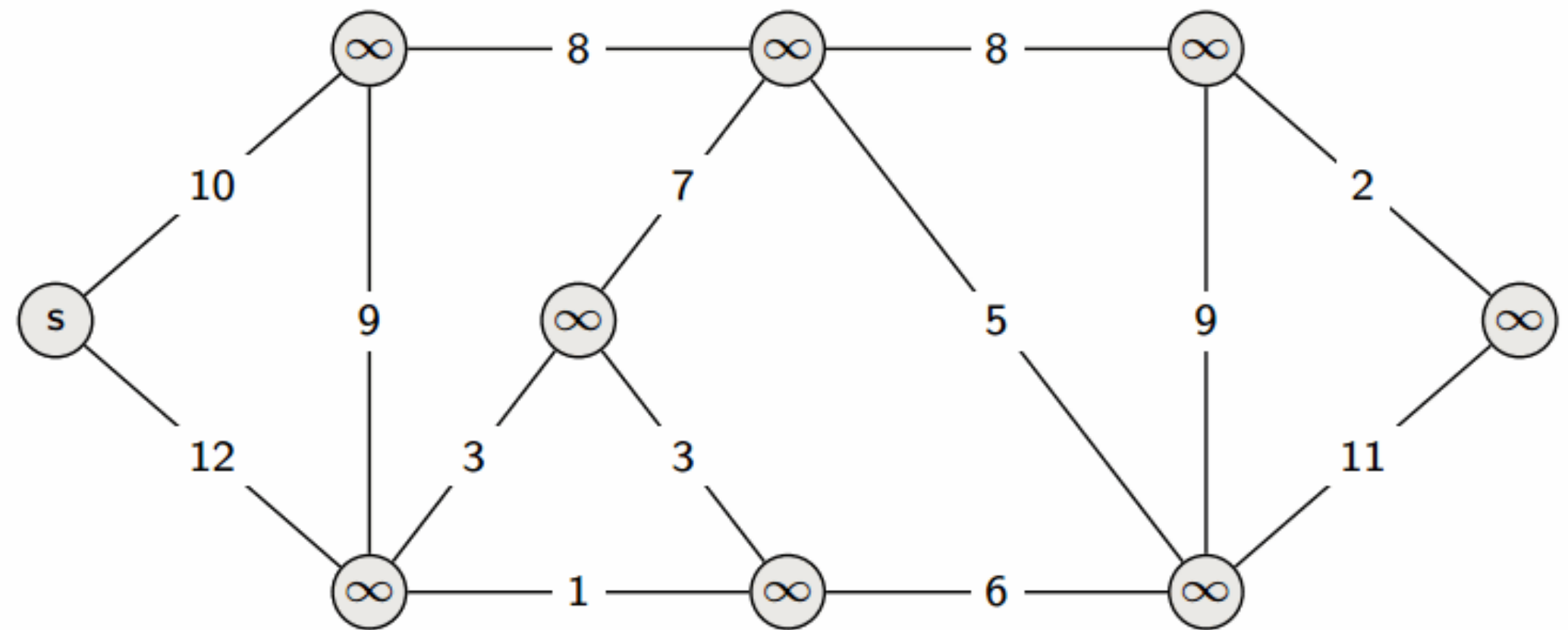


# Графы 2



# Алгоритм Дейкстры

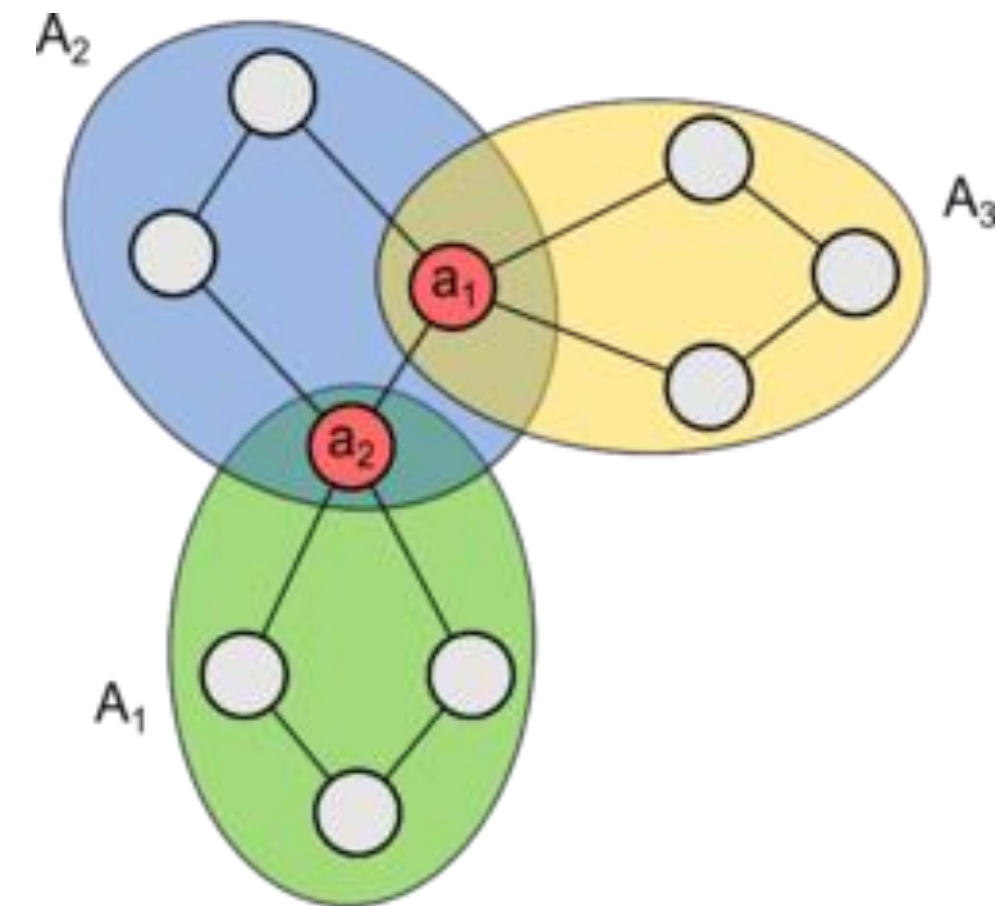
- Положим во все вершины  $+\infty$ , а в начальную 0
- Будем находить непосещенную вершину с самым малым значением и отмечать ее как посещенную
- Затем для обновляем расстояния для всех смежных вершин



# Запуск из нескольких вершин

Если очень захотеть, то алгоритм Дейкстры можно запустить из нескольких вершин.

- Как это сделать?
- Зачем нам это нужно?
- Как изменится сложность?

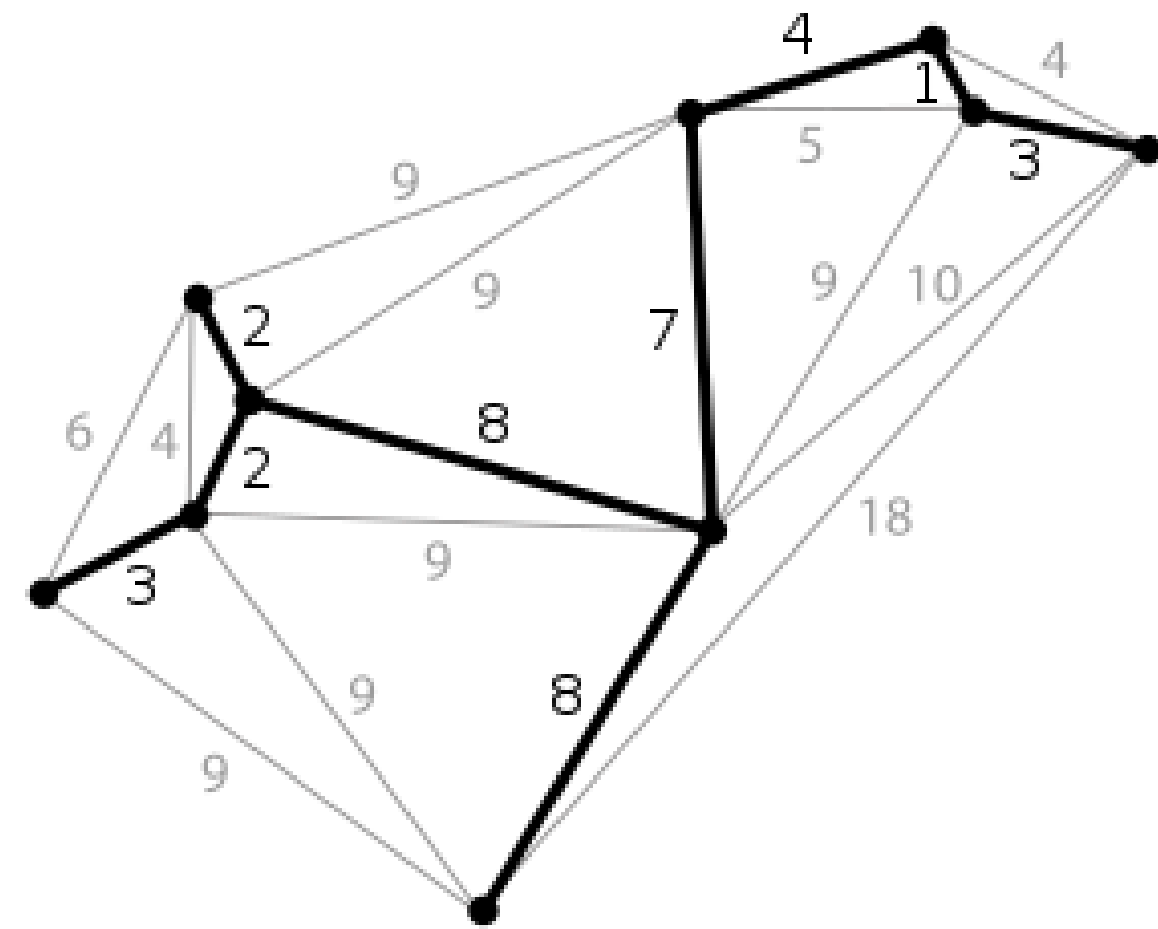


# Минимальные остов

Есть граф, хотим из ребер графа  
построить минимальное по весу  
дерево

Для этого нам нужна лемма о безопасном ребре:

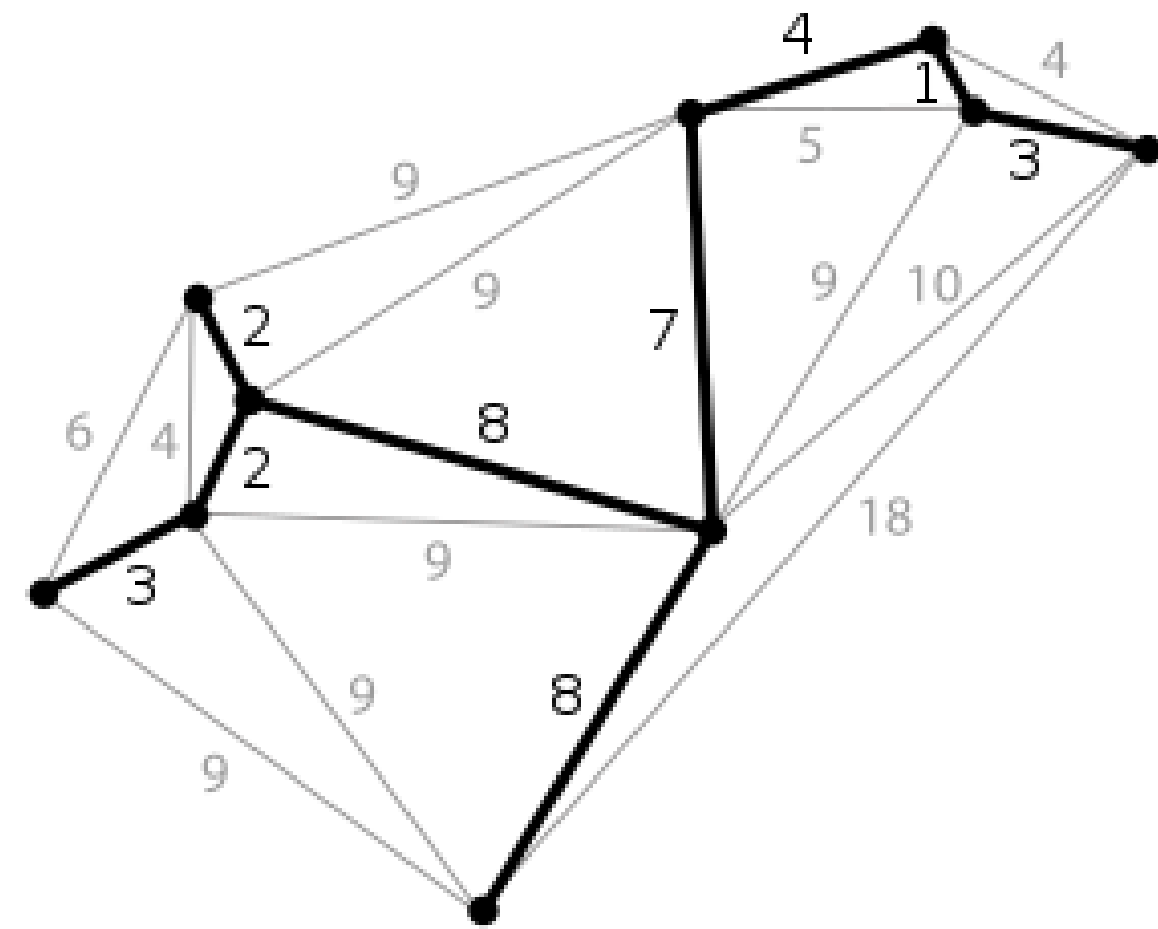
Если разделить граф на две части,  
то объединить их оптимальным  
образом можно ребром  
минимального веса между двумя  
частями\*



# Алгоритм Прима

Ищем минимальное исходящее ребро из нашего остова, и добавляем его в остов

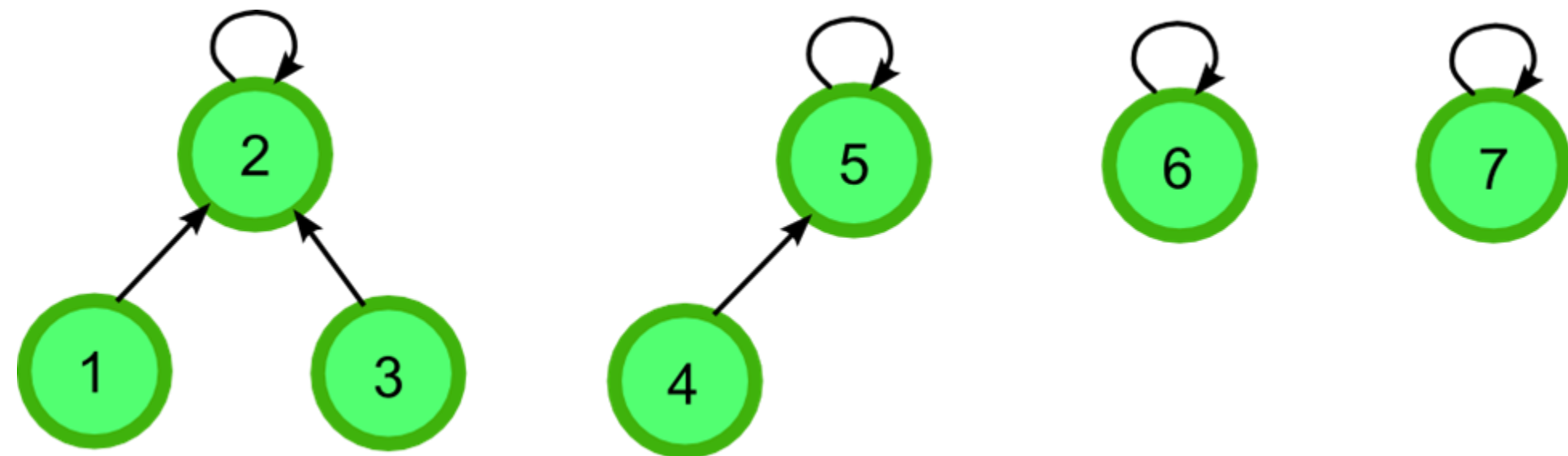
- Какая сложность будет у этого алгоритма?
- Похоже на Дейкстру?



# Система непересекающихся множеств

Хотим объединять множества. К примеру, компоненты связности в графе

- Изначально все компоненты будут одни в своей компоненте, потом мы будем их объединять друг с другом
- Структура похоже на дерево, и нам важно знать корень каждой из компонент



# Как объединять множества быстрее?



Эвристика сжатия пути

Будем переподвешивать  
все на пути до корня



Весовая эвристика

Будем подвешивать  
меньшие деревья за  
большие



А что если будем  
делать и то, и то?

# Как объединять множества быстрее?



Сидите с пацанами  
кайфуете

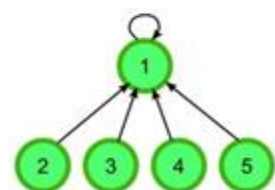


и тут стасян такой



пацаны смотрите че  
умею

оп



Бля стасян ну ты и  
дебил конечно



Весовая эвристика

Будем подвешивать  
меньшие деревья за  
большие



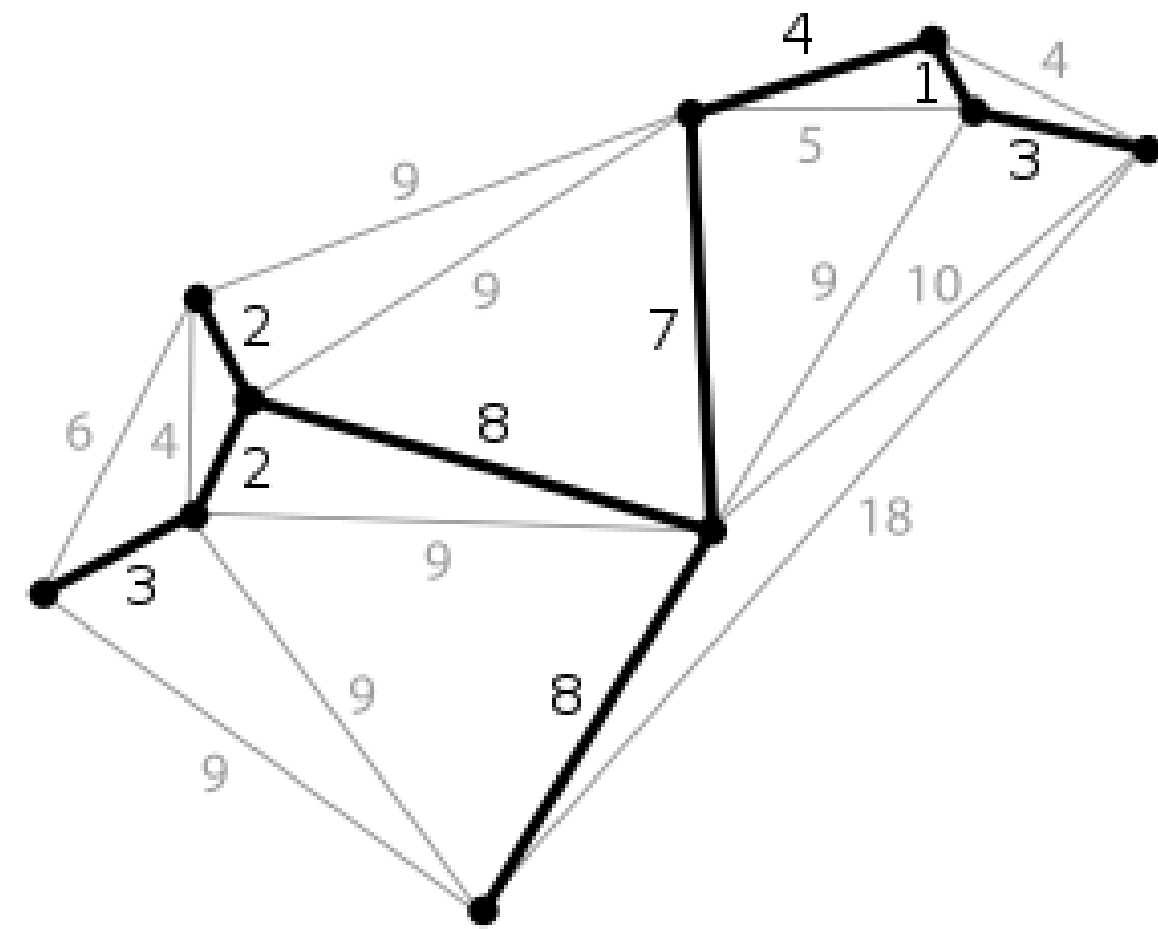
А что если будем  
делать и то, и то?



# Алгоритм Краскала

Сортируем все ребра и берем их в  
остов, если ребра соединяют  
вершины из разных компонент

- Какая сложность будет у этого алгоритма?
- Как нам поможет СНМ?





**Спасибо!**

