

**Переборные задачи. NP.  
Поиск с возвратом.  
Метод ветвей и границ.**

И снова этот C++

# NP-complete. Суть без теории.

Такой тип задач, точное решение которых мы находим **полным или частичным перебором** всех возможных решений.

Трудоёмкость таких задач — **экспоненциальная**.

Проверка ответа при требует меньшей трудоёмкости.

Почему так?

$$P \neq NP$$

(наверное)

# NP-complete. Суть без теории.

Такой тип задач, точное решение которых мы находим **полным или частичным перебором** всех возможных решений.

Трудоёмкость таких задач — **экспоненциальная**.

Проверка ответа при требует меньшей трудоёмкости.

Почему так?

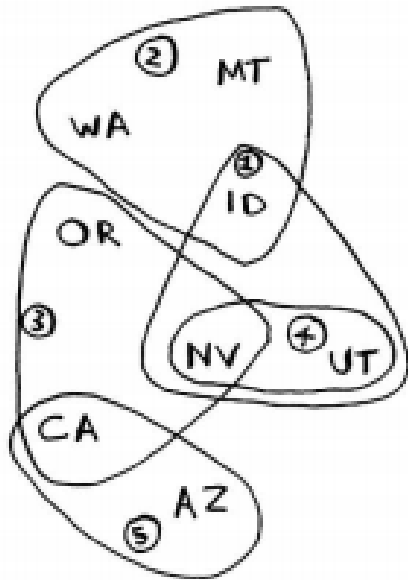
# Типичные NP-C задачи

- Задача о выполнимости булевых формул
- Проблема раскраски графа
- Задача коммивояжёра
- Задача о покрытии множества
- Задача о клике

# Задача о покрытии множества

Каждая радиостанция покрывает один или несколько штатов.

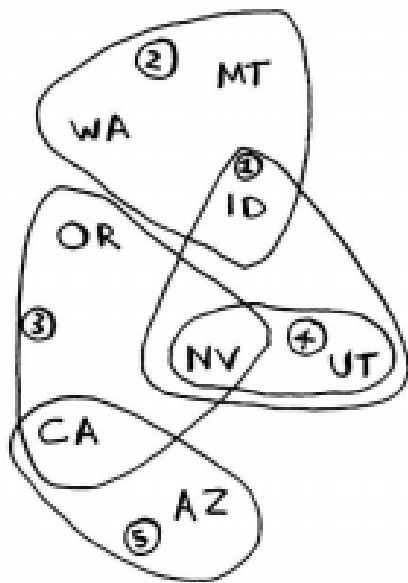
**Требуется:** найти минимальный набор радиостанций, покрывающий все штаты.



# Задача о покрытии множества

Каждая радиостанция покрывает один или несколько штатов.

**Требуется:** найти минимальный набор радиостанций, покрывающий все штаты.



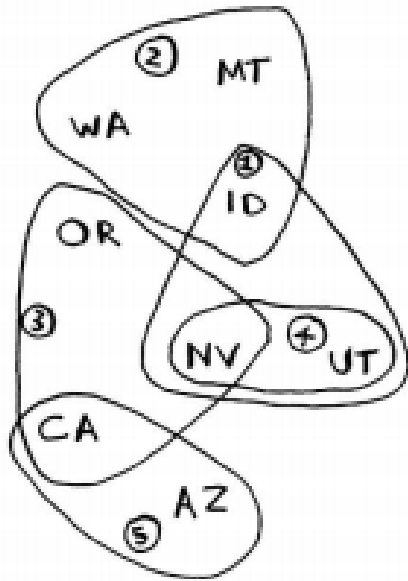
**Решение:** генерируем все возможные подмножества станций. Выбираем из них те, которые покрывают все штаты. Находим среди них наименьший.

**В чём подвох?**

# Задача о покрытии множества

Каждая радиостанция покрывает один или несколько штатов.

**Требуется:** найти минимальный набор радиостанций, покрывающий все штаты.



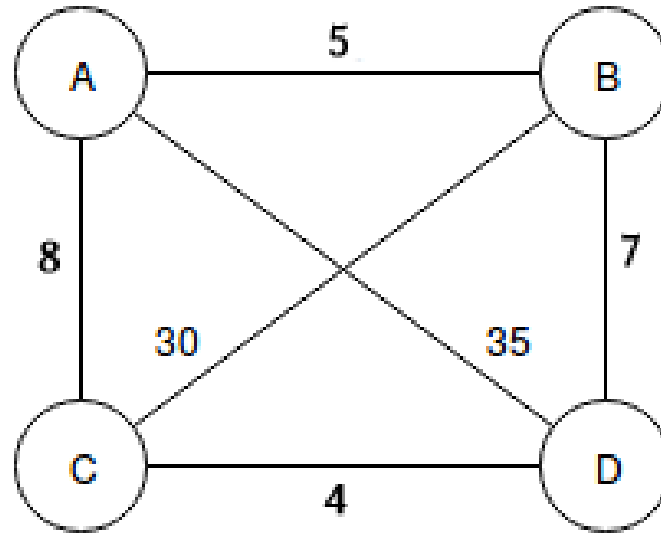
**Решение:** генерируем все возможные подмножества станций. Выбираем из них те, которые покрывают все штаты. Находим среди них наименьший.

**В чём подвох?**

Размер такого множества —  $2^N$ .

# Задача коммивояжёра. Метод ветвей и границ.

**Требуется:** найти минимальный путь, посещающий все города хотя бы по одному разу.

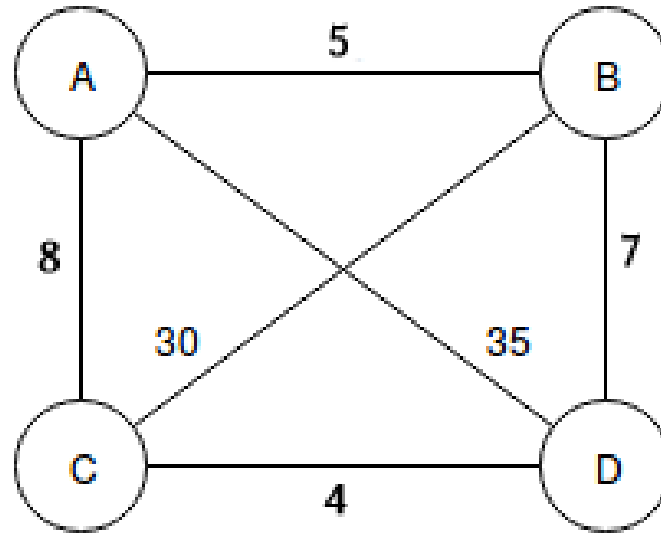


**Суть:** полный перебор всех возможных вариантов с отсеиванием явно неоптимальных решений.



# Задача коммивояжёра. Метод ветвей и границ.

**Требуется:** найти минимальный путь, посещающий все города хотя бы по одному разу.



# Поиск с возвратом.

**Прямой ход:** вынесение отдельной подзадачи, её решение.

**Обратный ход:** оптимизируем решение, очищаем стек.