# Базовые подсчеты

Владимир Подольский

Факультет компьютерных наук, Высшая Школа Экономики

# Базовые подсчеты

Зачем изучать подсчеты?

Правило суммы

Удобный язык: множества

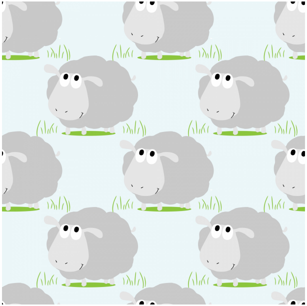
Обобщение правила суммы

Рекурсивные подсчеты: число путей

Правило произведения

# Зачем изучать подсчеты?

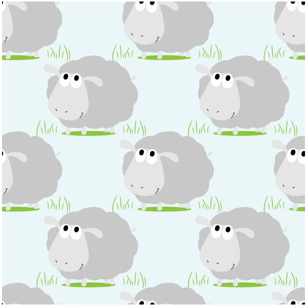
- Подсчет — одна из самых базовых задач, связанных с математикой



Needpix.com

# Зачем изучать подсчеты?

- Подсчет — одна из самых базовых задач, связанных с математикой
- Цель: сказать, сколько у нас есть объектов, не пересчитывая их один за другим



Needpix.com

# Зачем изучать подсчеты?

- Подсчеты используются во многих разделах математики и приложениях

# Зачем изучать подсчеты?

- Подсчеты используются во многих разделах математики и приложениях
- Важное приложение: подсчет числа шагов работы алгоритма

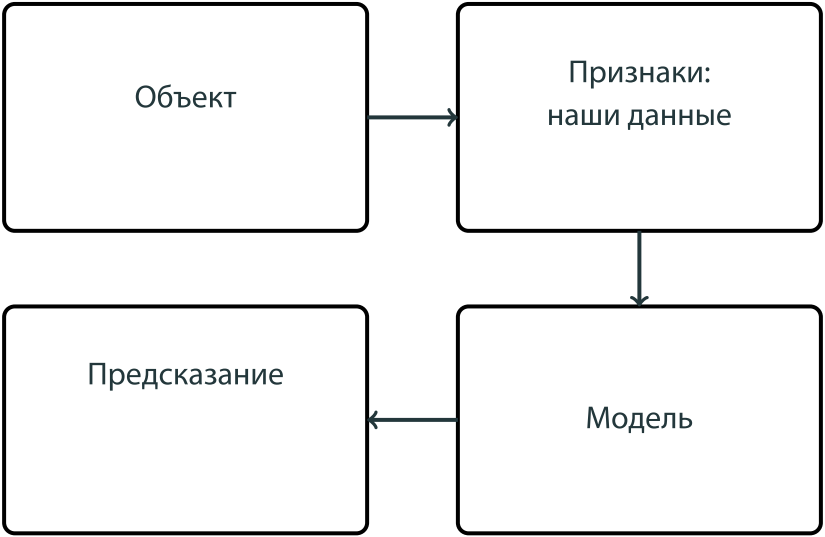
# Зачем изучать подсчеты?

- Подсчеты используются во многих разделах математики и приложениях
- Важное приложение: подсчет числа шагов работы алгоритма
- Важное приложение: подсчет вероятностей

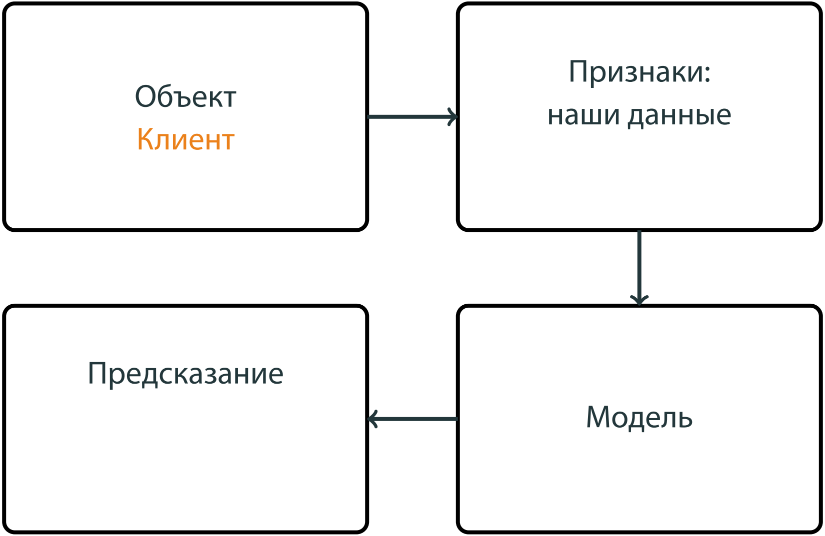
# Зачем изучать подсчеты?

- Подсчеты используются во многих разделах математики и приложениях
- Важное приложение: подсчет числа шагов работы алгоритма
- Важное приложение: подсчет вероятностей
- Важное приложение: оценка количества данных

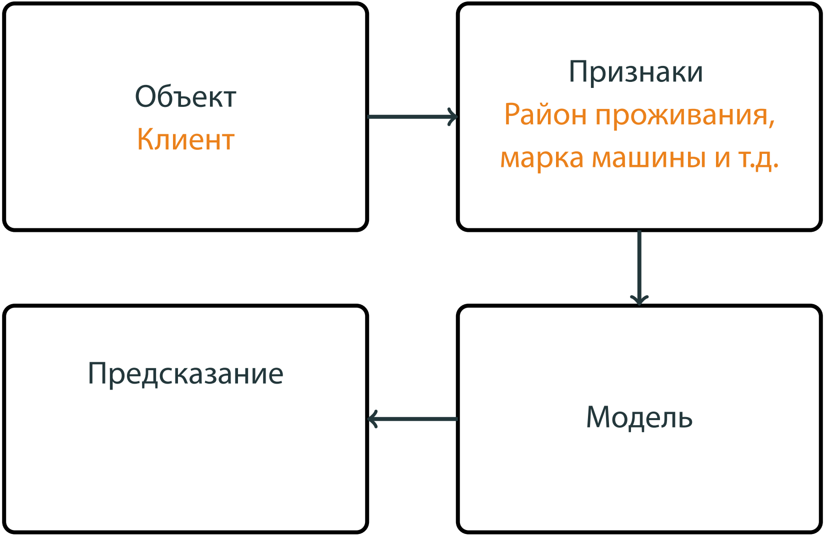
# Подсчеты в анализе данных



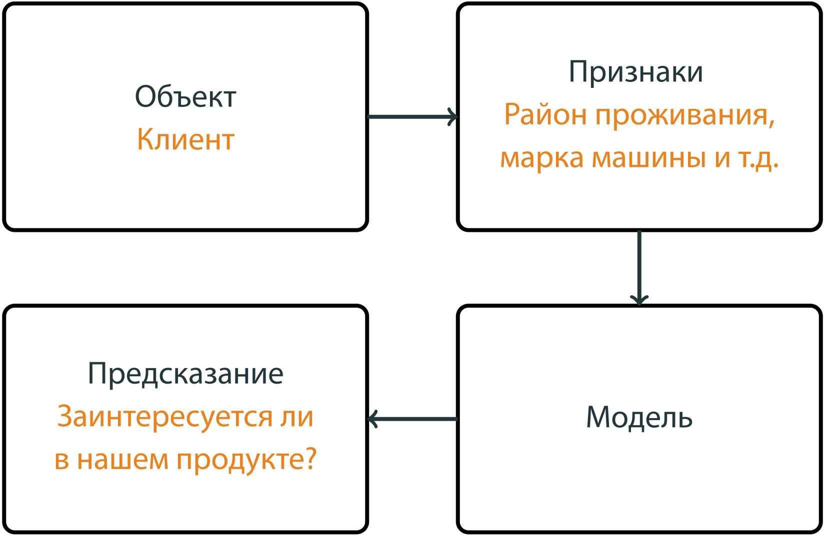
# Подсчеты в анализе данных



# Подсчеты в анализе данных



# Подсчеты в анализе данных



# Подсчеты в анализе данных

- Прежде чем использовать модель нужно ее обучить

# Подсчеты в анализе данных

- Прежде чем использовать модель нужно ее обучить
- Какое количество данных потребуется для обучения?

# Подсчеты в анализе данных

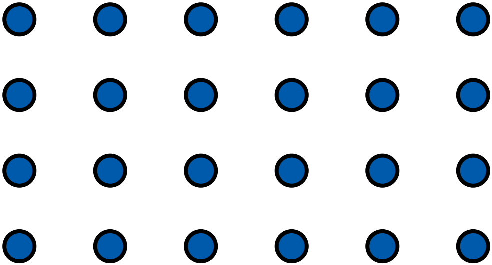
- Прежде чем использовать модель нужно ее обучить
- Какое количество данных потребуется для обучения?
- Во многих моделях количество данных, необходимых для обучения, сопоставимо с количеством всех возможных объектов в нашем пространстве признаков

# Подсчеты в анализе данных

- Прежде чем использовать модель нужно ее обучить
- Какое количество данных потребуется для обучения?
- Во многих моделях количество данных, необходимых для обучения, сопоставимо с количеством всех возможных объектов в нашем пространстве признаков
- Полезно оценить количеством всех возможных совокупностей значений наших признаков

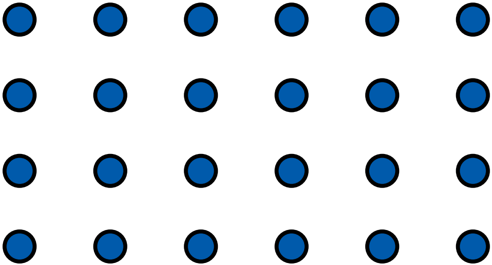
# Зачем изучать подсчеты?

- С другой стороны, есть простые и важные идеи, которые помогают в подсчетах



# Зачем изучать подсчеты?

- С другой стороны, есть простые и важные идеи, которые помогают в подсчетах
- Игрушечный пример: можете ли вы сказать, сколько кругов на рисунке, не подсчитывая их один за другим?



# Пример из жизни, анонс


wikimedia.org

- Предположим, что мы хотим ввести новый формат автомобильных номеров

# Пример из жизни, анонс


wikimedia.org

- Предположим, что мы хотим ввести новый формат автомобильных номеров
- Российский номер: 3 цифры, 3 буквы; 78 — код региона

# Пример из жизни, анонс


wikimedia.org

- Предположим, что мы хотим ввести новый формат автомобильных номеров
- Российский номер: 3 цифры, 3 буквы; 78 — код региона
- У нас есть: 10 вариантов для цифр, 12 вариантов для букв (используются только буквы, у которых есть аналог в латинице)

# Пример из жизни, анонс


wikimedia.org

- Предположим, что мы хотим ввести новый формат автомобильных номеров
- Российский номер: 3 цифры, 3 буквы; 78 — код региона
- У нас есть: 10 вариантов для цифр, 12 вариантов для букв (используются только буквы, у которых есть аналог в латинице)
- Хватит ли нам автомобильных номеров для всех?

# Что мы узнали

- Подсчеты — важная и нужная задача

# Что мы узнали

- Подсчеты — важная и нужная задача
- При этом есть разные приемы того, как эффективно подсчитывать объекты

# Что мы узнали

- Подсчеты — важная и нужная задача
- При этом есть разные приемы того, как эффективно подсчитывать объекты
- Их мы и обсудим в ближайшие две недели

# Что мы узнали

- Подсчеты — важная и нужная задача
- При этом есть разные приемы того, как эффективно подсчитывать объекты
- Их мы и обсудим в ближайшие две недели
- На третьей неделе мы применим эти знания к подсчету вероятностей

# Базовые подсчеты

Зачем изучать подсчеты?

Правило суммы

Удобный язык: множества

Обобщение правила суммы

Рекурсивные подсчеты: число путей

Правило произведения

# Правило суммы

**Правило суммы**

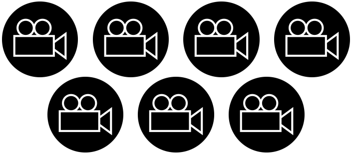
Если у нас $k$ объектов первого типа и $n$ объектов второго типа, то у нас есть $n + k$ объектов одного из двух типов

# Правило суммы

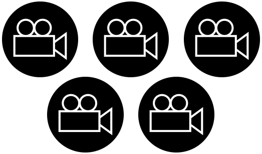
**Правило суммы**

Если у нас $k$ объектов первого типа и $n$ объектов второго типа, то у нас есть $n + k$ объектов одного из двух типов

Видео $\geq$ 10 мин.



Видео $<$ 10 мин.

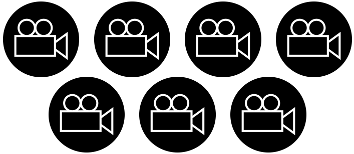


# Правило суммы

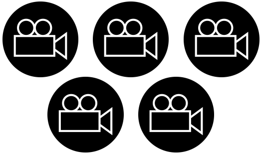
**Правило суммы**

Если у нас $k$ объектов первого типа и $n$ объектов второго типа, то у нас есть $n + k$ объектов одного из двух типов

Видео $\geq$ 10 мин.



Видео $<$ 10 мин.

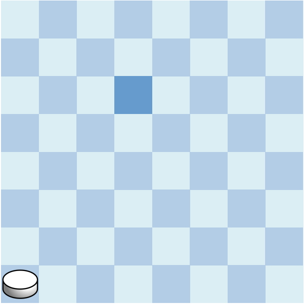


всего 7+5=12 видео

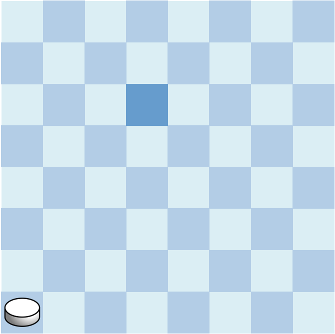
# Пример: фишка на доске

## Фишка на доске

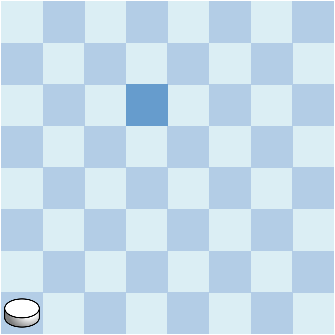
Фишка стоит в левой нижней клетке шахматной доски. За один ход ее можно переместить на одно поле направо или вверх. Сколько ходов нужно сделать, чтобы попасть в клетку, отмеченную на рисунке?



# Пример: фишка на доске

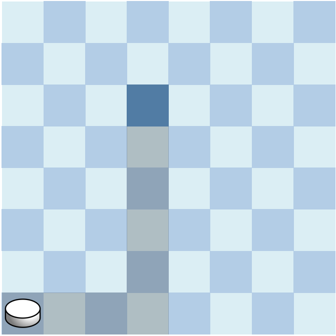


# Пример: фишка на доске



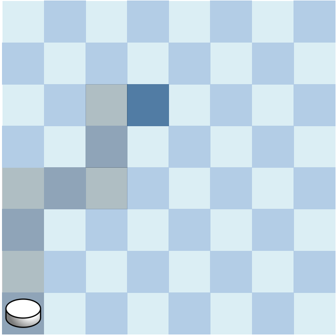
- Мы можем передвинуть фишку несколькими способами

# Пример: фишка на доске



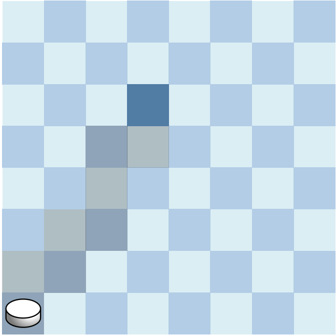
- Мы можем передвинуть фишку несколькими способами

# Пример: фишка на доске



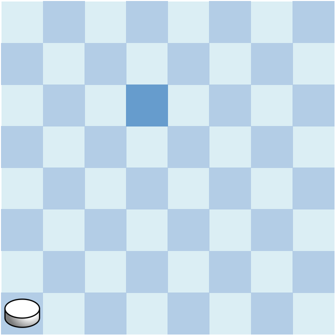
- Мы можем передвинуть фишку несколькими способами

# Пример: фишка на доске



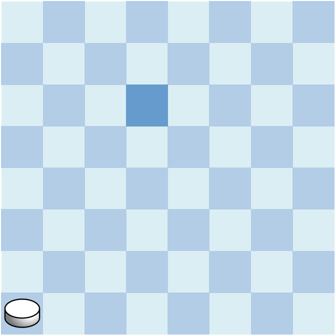
- Мы можем передвинуть фишку несколькими способами

# Пример: фишка на доске



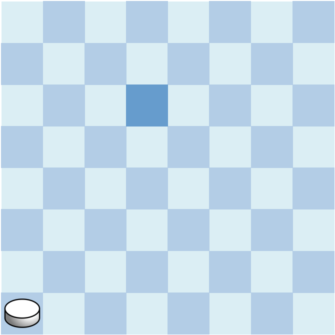
- Мы можем передвинуть фишку несколькими способами
- Во всех случаях нужно 8 ходов

# Пример: фишка на доске



- Мы можем передвинуть фишку несколькими способами
- Во всех случаях нужно 8 ходов
- И это не случайно

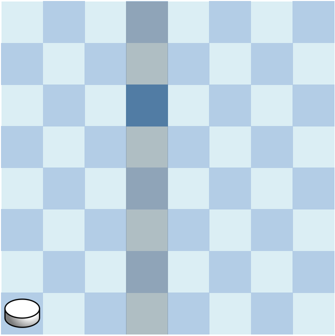
# Пример: фишка на доске



- Мы можем передвинуть фишку несколькими способами
- Во всех случаях нужно 8 ходов
- И это не случайно

1. Есть два типа ходов: ходы вправо и ходы вверх

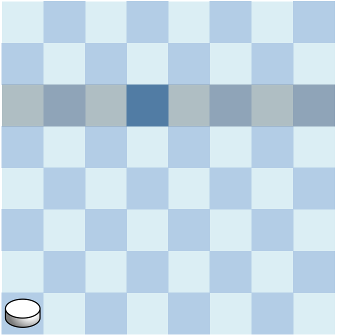
# Пример: фишка на доске



- Мы можем передвинуть фишку несколькими способами
- Во всех случаях нужно 8 ходов
- И это не случайно

1. Есть два типа ходов: ходы вправо и ходы вверх
2. Чтобы попасть в 4-й столбец нужно 3 хода вправо

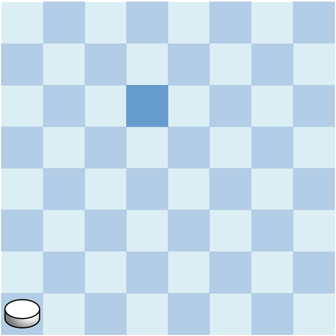
# Пример: фишка на доске



- Мы можем передвинуть фишку несколькими способами
- Во всех случаях нужно 8 ходов
- И это не случайно

1. Есть два типа ходов: ходы вправо и ходы вверх
2. Чтобы попасть в 4-й столбец нужно 3 хода вправо
3. Чтобы попасть в 6-ю строку нужно 5 ходов вверх

# Пример: фишка на доске



- Мы можем передвинуть фишку несколькими способами
- Во всех случаях нужно 8 ходов
- И это не случайно

1. Есть два типа ходов: ходы вправо и ходы вверх
2. Чтобы попасть в 4-й столбец нужно 3 хода вправо
3. Чтобы попасть в 6-ю строку нужно 5 ходов вверх
4. Всего нужно 3+5=8 ходов

# Еще один пример

- Пусть в наших данных у нас есть 7 видео длиной $\geq 4$ минуты и 5 музыкальных клипов

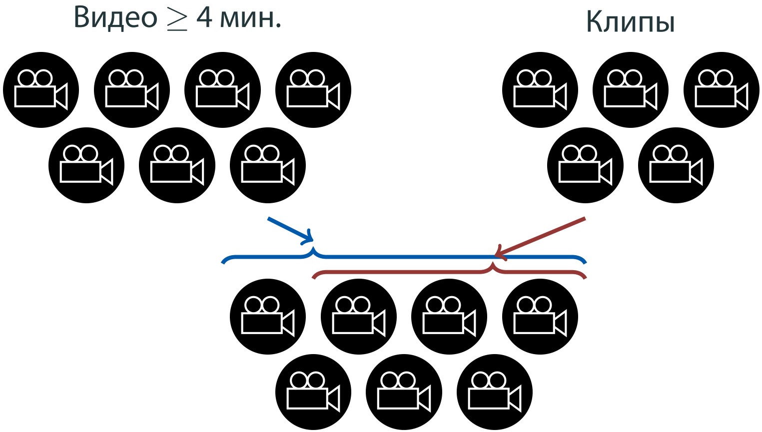
# Еще один пример

- Пусть в наших данных у нас есть 7 видео длиной $\geq 4$ минуты и 5 музыкальных клипов
- Можно ли утверждать, что всего у нас 12 видео, которые длиной $\geq 4$ минуты или клипы?

# Еще один пример

- Пусть в наших данных у нас есть 7 видео длиной $\geq 4$ минуты и 5 музыкальных клипов
- Можно ли утверждать, что всего у нас 12 видео, которые длиной $\geq 4$ минуты или клипы?
- Что если некоторые клипы длиннее 4 минут?

# Еще один пример




Правило суммы в этом примере не работает!

# Чуть более хитрый пример

- Пусть наш видеосервис сортирует видео по категориям: музыка, смешные видео, садоводство, животные и т.д.

# Чуть более хитрый пример

- Пусть наш видеосервис сортирует видео по категориям: музыка, смешные видео, садоводство, животные и т.д.
- Пусть в наших данных собраны 7 видео из одной категории и 5 видео из другой

# Чуть более хитрый пример

- Пусть наш видеосервис сортирует видео по категориям: музыка, смешные видео, садоводство, животные и т.д.
- Пусть в наших данных собраны 7 видео из одной категории и 5 видео из другой
- Верно ли, что у нас всего 12 видео, попадающих в одну из двух этих категорий?

# Чуть более хитрый пример

- Пусть наш видеосервис сортирует видео по категориям: музыка, смешные видео, садоводство, животные и т.д.
- Пусть в наших данных собраны 7 видео из одной категории и 5 видео из другой
- Верно ли, что у нас всего 12 видео, попадающих в одну из двух этих категорий?
- Не обязательно! Категории могут пересекаться

# Чуть более хитрый пример

- Пусть наш видеосервис сортирует видео по категориям: музыка, смешные видео, садоводство, животные и т.д.
- Пусть в наших данных собраны 7 видео из одной категории и 5 видео из другой
- Верно ли, что у нас всего 12 видео, попадающих в одну из двух этих категорий?
- Не обязательно! Категории могут пересекаться
- Например, у нас могут быть смешные видео с животными

# Еще раз о правиле суммы

**Правило суммы**

Если у нас $k$ объектов первого типа и $n$ объектов второго типа, то у нас есть $n + k$ объектов одного из двух типов

- Важный урок: в правиле суммы никакие объекты не должны принадлежать сразу обоим классам!

# Базовые подсчеты

Зачем изучать подсчеты?

Правило суммы

Удобный язык: множества

Обобщение правила суммы

Рекурсивные подсчеты: число путей

Правило произведения

# Удобный язык: множества

- Множеством называется произвольная совокупность объектов

# Удобный язык: множества

- Множеством называется произвольная совокупность объектов
- Будем обозначать множества заглавными буквами: $A$, $B$, $C$, $S$ и т.д.

# Удобный язык: множества

- Множеством называется произвольная совокупность объектов
- Будем обозначать множества заглавными буквами: $A$, $B$, $C$, $S$ и т.д.
- Множества можно задавать списком их элементов: $S = \{0, 1, 2, 3\}$; это множество состоит из 4 элементов: $0$, $1$, $2$, $3$

# Удобный язык: множества

- Множеством называется произвольная совокупность объектов
- Будем обозначать множества заглавными буквами: $A$, $B$, $C$, $S$ и т.д.
- Множества можно задавать списком их элементов: $S = \{0, 1, 2, 3\}$; это множество состоит из 4 элементов: $0$, $1$, $2$, $3$
- Порядок элементов не важен: $\{0, 1, 2, 3\} = \{2, 0, 3, 1\}$

# Удобный язык: множества

- Множеством называется произвольная совокупность объектов
- Будем обозначать множества заглавными буквами: $A$, $B$, $C$, $S$ и т.д.
- Множества можно задавать списком их элементов: $S = \{0, 1, 2, 3\}$; это множество состоит из 4 элементов: $0$, $1$, $2$, $3$
- Порядок элементов не важен: $\{0, 1, 2, 3\} = \{2, 0, 3, 1\}$
- Повторы в списке элементов не важны: $\{0, 1, 2, 3\} = \{1, 0, 1, 3, 2, 3\}$

# Удобный язык: множества

- Нам множества дают удобный язык

# Удобный язык: множества

- Нам множества дают удобный язык
- В математике множества играют важную и ключевую роль

# Удобный язык: множества

- Нам множества дают удобный язык
- В математике множества играют важную и ключевую роль
- Для нас множества состоят из чего угодно:
  $S = \{0, \sqrt{2}, \text{ Исаак Ньютон}, \text{ русалка}\}$

# Удобный язык: множества

- Нам множества дают удобный язык
- В математике множества играют важную и ключевую роль
- Для нас множества состоят из чего угодно: $S = \{0, \sqrt{2}, \text{ Исаак Ньютон}, \text{ русалка}\}$
- Но здесь есть тонкости

# Удобный язык: множества

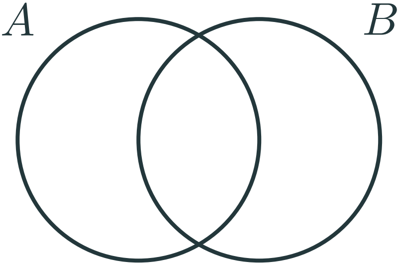
- Нам множества дают удобный язык
- В математике множества играют важную и ключевую роль
- Для нас множества состоят из чего угодно: $S = \{0, \sqrt{2}, \text{ Исаак Ньютон}, \text{ русалка}\}$
- Но здесь есть тонкости
- “Множество, состоящее из всех множеств” — опасная конструкция

# Удобный язык: множества

- Нам множества дают удобный язык
- В математике множества играют важную и ключевую роль
- Для нас множества состоят из чего угодно: $S = \{0, \sqrt{2}, \text{ Исаак Ньютон}, \text{ русалка}\}$
- Но здесь есть тонкости
- “Множество, состоящее из всех множеств” — опасная конструкция
- В наших курсах мы не столкнемся с такими трудностями и не будем их обсуждать

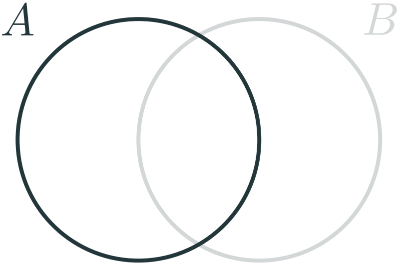
# Диаграммы Венна

- Множества удобно изображать с помощью диаграмм Венна



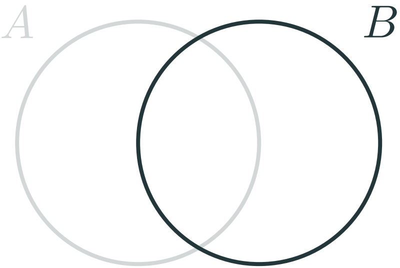
# Диаграммы Венна

- Множества удобно изображать с помощью диаграмм Венна
- Элементы $A$ лежат в левом круге



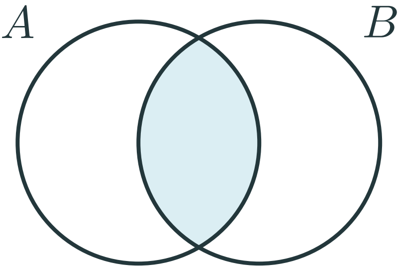
# Диаграммы Венна

- Множества удобно изображать с помощью диаграмм Венна
- Элементы $A$ лежат в левом круге
- Элементы $B$ лежат в правом круге

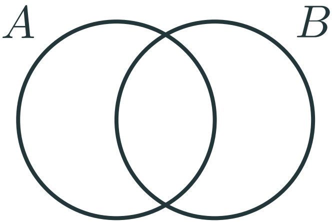


# Диаграммы Венна

- Множества удобно изображать с помощью диаграмм Венна
- Элементы $A$ лежат в левом круге
- Элементы $B$ лежат в правом круге
- В пересечении лежат элементы, принадлежащие обоим множествам

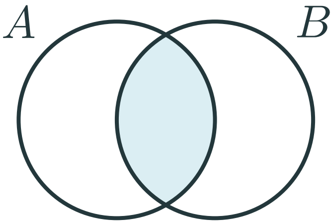


# Удобные обозначения



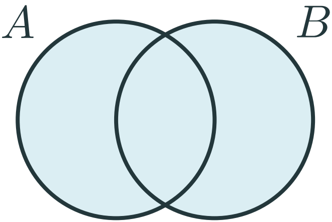

- Пусть у нас есть два множества $A$ и $B$

# Удобные обозначения



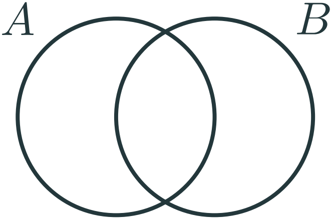

- Пусть у нас есть два множества $A$ и $B$
- Множество $A \cap B$ называется пересечением $A$ и $B$: оно состоит из элементов, лежащих в обоих множествах

# Удобные обозначения



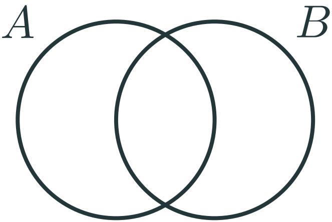

- Пусть у нас есть два множества $A$ и $B$
- Множество $A \cap B$ называется пересечением $A$ и $B$: оно состоит из элементов, лежащих в обоих множествах
- Множество $A \cup B$ называется объединением $A$ и $B$: оно состоит из элементов, лежащих хотя бы в одном из двух множеств

# Удобные обозначения



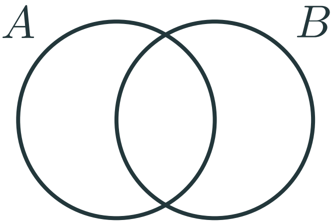

- Если всякий элемент $A$ является также элементом $B$, то $A$ называется **подмножеством** $B$; мы пишем $A \subseteq B$

# Удобные обозначения



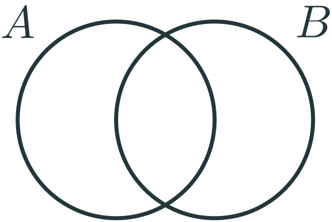

- Если всякий элемент $A$ является также элементом $B$, то $A$ называется подмножеством $B$; мы пишем $A \subseteq B$
- Мы говорим, что $A$ вложено в $B$

# Удобные обозначения



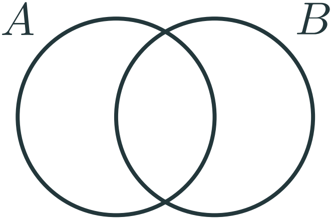

- Если всякий элемент $A$ является также элементом $B$, то $A$ называется подмножеством $B$; мы пишем $A \subseteq B$
- Мы говорим, что $A$ вложено в $B$
- Если $x$ является элементом $A$, мы пишем $x \in A$

# Удобные обозначения



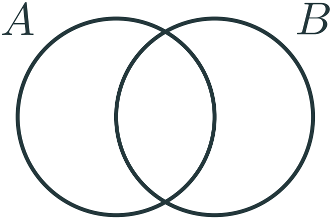

- Если всякий элемент $A$ является также элементом $B$, то $A$ называется подмножеством $B$; мы пишем $A \subseteq B$
- Мы говорим, что $A$ вложено в $B$
- Если $x$ является элементом $A$, мы пишем $x \in A$
- Мы говорим, что $x$ принадлежит $A$

# Удобные обозначения




- Количество элементов множества $A$ обозначается через $|A|$ (может быть бесконечным)

# Удобные обозначения




- Количество элементов множества $A$ обозначается через $|A|$ (может быть бесконечным)
- Множество без элементов обозначается через $\emptyset$ и называется пустым множеством

# Базовые подсчеты

Зачем изучать подсчеты?

Правило суммы

Удобный язык: множества

Обобщение правила суммы

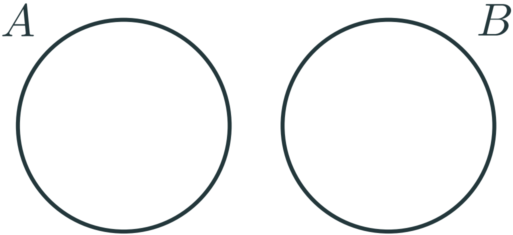
Рекурсивные подсчеты: число путей

Правило произведения

# Правило суммы на языке множеств

## Правило суммы

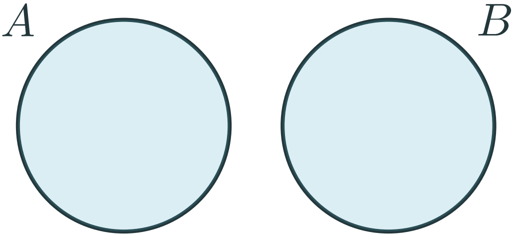
Если множество $A$ содержит $k$ элементов, а множество $B$ — $n$ элементов и эти множества не содержат общих элементов, то множество $A \cup B$ содержит $n + k$ элементов



# Правило суммы на языке множеств

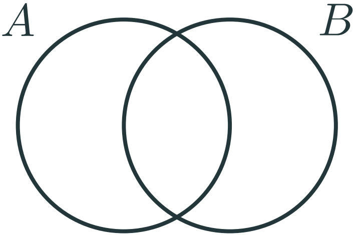
**Правило суммы**

Если множество $A$ содержит $k$ элементов, а множество $B$ — $n$ элементов и эти множества не содержат общих элементов, то множество $A \cup B$ содержит $n + k$ элементов



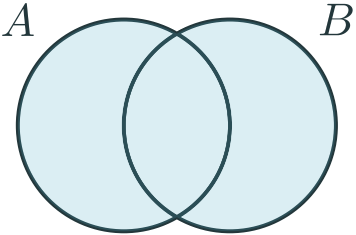
# Обобщенное правило суммы

Но что если мы хотим посчитать $|A \cup B|$ в ситуации на картинке?

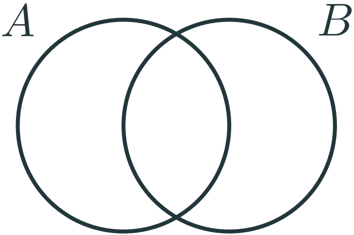


# Обобщенное правило суммы

Но что если мы хотим посчитать $|A \cup B|$ в ситуации на картинке?

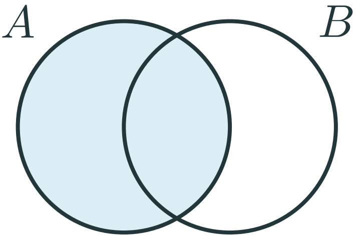


# Обобщенное правило суммы



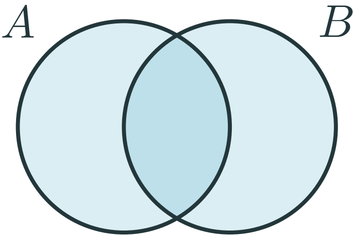

- Если мы рассмотрим $|A| + |B|$ как в правиле суммы, это не будет правильным ответом

# Обобщенное правило суммы



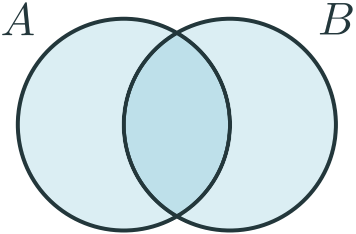

- Если мы рассмотрим $|A| + |B|$ как в правиле суммы, это не будет правильным ответом

# Обобщенное правило суммы



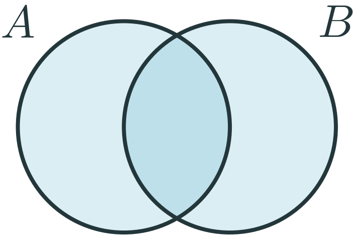

- Если мы рассмотрим $|A| + |B|$ как в правиле суммы, это не будет правильным ответом

# Обобщенное правило суммы



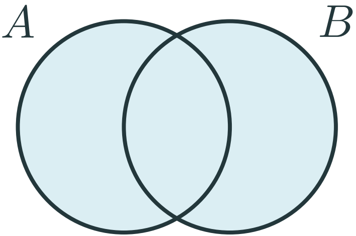

- Если мы рассмотрим $|A| + |B|$ как в правиле суммы, это не будет правильным ответом
- Мы считаем элементы, лежащие в обоих множествах $A$ и $B$ дважды

# Обобщенное правило суммы



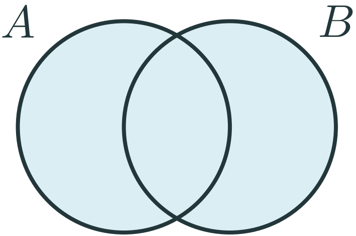

- Если мы рассмотрим $|A| + |B|$ как в правиле суммы, это не будет правильным ответом
- Мы считаем элементы, лежащие в обоих множествах $A$ и $B$ дважды
- Тогда давайте их вычтем!

# Обобщенное правило суммы




- Если мы рассмотрим $|A| + |B|$ как в правиле суммы, это не будет правильным ответом
- Мы считаем элементы, лежащие в обоих множествах $A$ и $B$ дважды
- Тогда давайте их вычтем!

# Обобщенное правило суммы



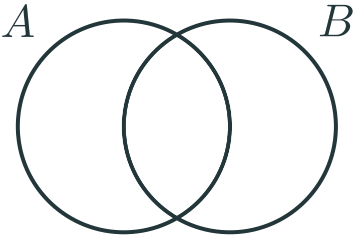

- Если мы рассмотрим $|A| + |B|$ как в правиле суммы, это не будет правильным ответом
- Мы считаем элементы, лежащие в обоих множествах $A$ и $B$ дважды
- Тогда давайте их вычтем!
- Это дает ответ: $|A \cup B| = |A| + |B| - |A \cap B|$

# Обобщенное правило суммы

## Правило суммы

Если множества $A$ и $B$ конечны, то

$$|A \cup B| = |A| + |B| - |A \cap B|$$



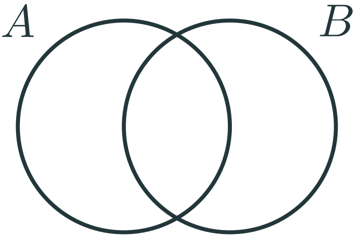
# Обобщенное правило суммы

## Правило суммы

Если множества $A$ и $B$ конечны, то

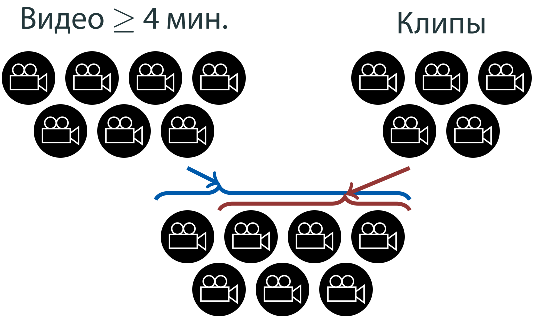
$$|A \cup B| = |A| + |B| - |A \cap B|$$

Покрывает обычное правило суммы: $|A \cap B| = 0$



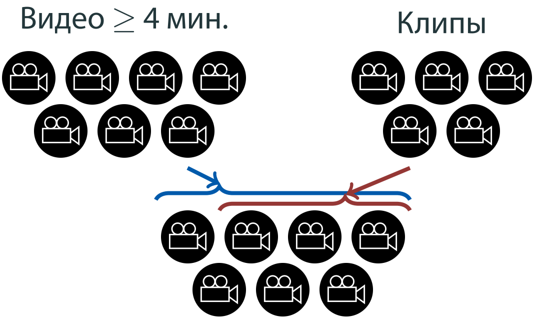
# Пример

Вернемся к примеру с видео



# Пример

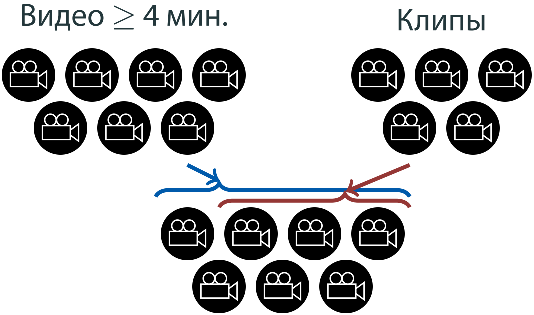
Вернемся к примеру с видео




- Размеры множеств здесь 7 и 5, размер пересечения — 5

# Пример

Вернемся к примеру с видео




- Размеры множеств здесь 7 и 5, размер пересечения — 5
- Получаем, что всего у нас $7 + 5 - 5 = 7$ видео

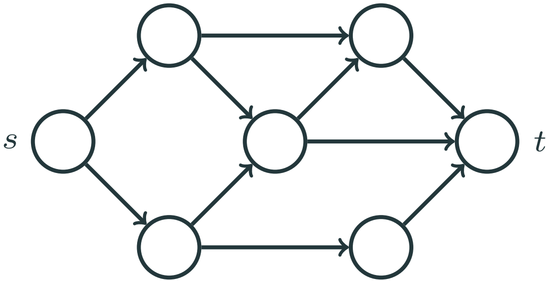
# Базовые подсчеты

- Зачем изучать подсчеты?
- Правило суммы
- Удобный язык: множества
- Обобщение правила суммы
- Рекурсивные подсчеты: число путей
- Правило произведения

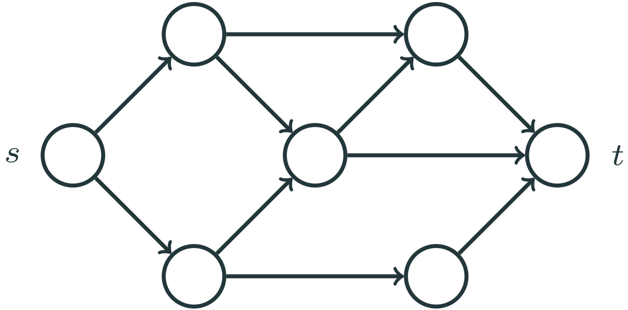
# Рекурсивные подсчеты: число путей

## Задача

Пусть у нас есть несколько точек, соединенных стрелками. Есть начальная точка $s$ (называемая источником) и конечная точка $t$ (называемая стоком). Сколько есть различных способов добраться из $s$ в $t$ по стрелкам?

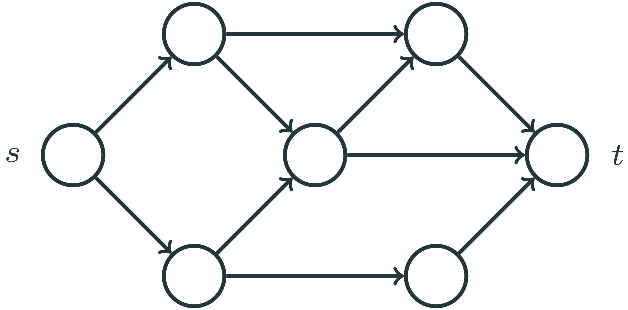


# Число путей



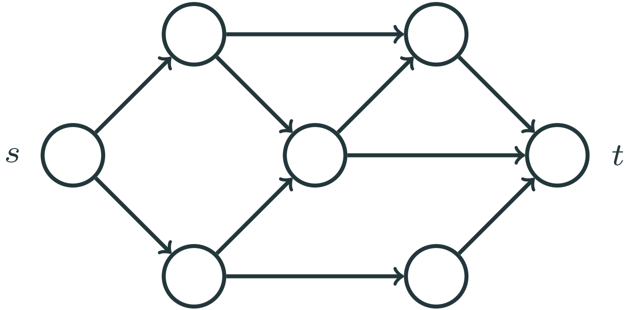

- Есть несколько разных путей; как посчитать их и ничего не пропустить?

# Число путей



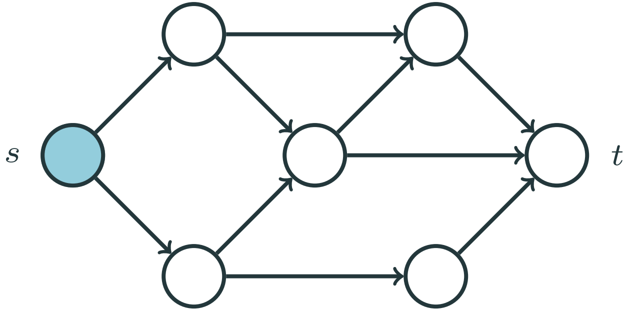

- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку

# Число путей



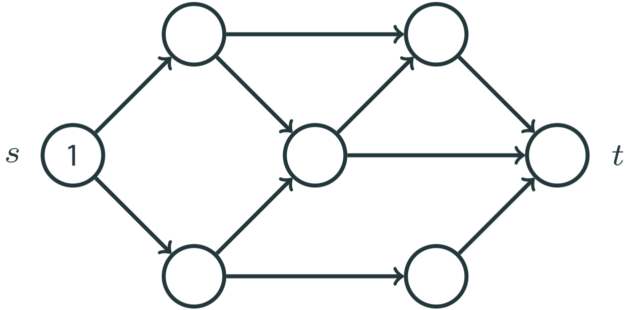

- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

# Число путей



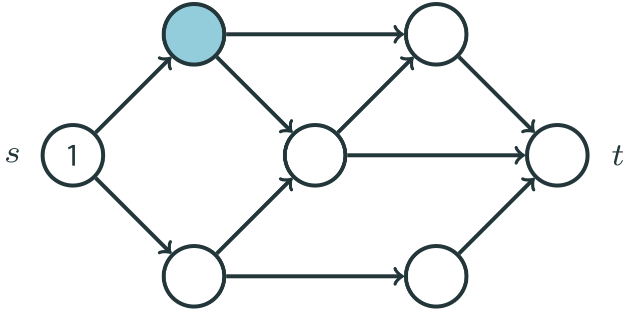

- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

# Число путей



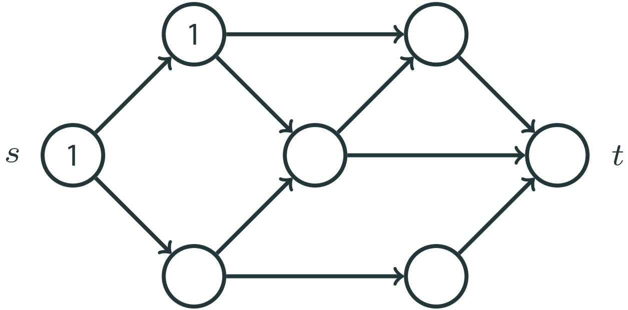

- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

# Число путей



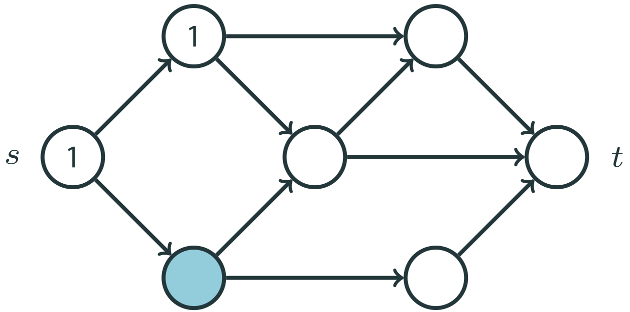

- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

# Число путей



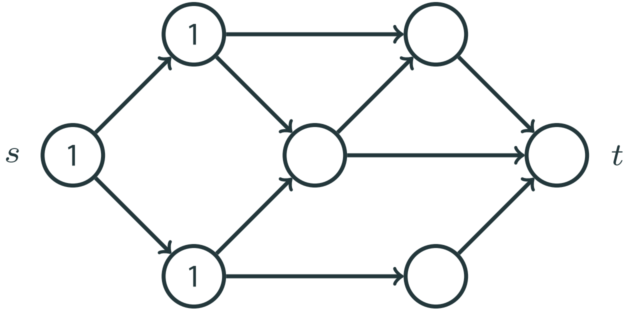

- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

# Число путей



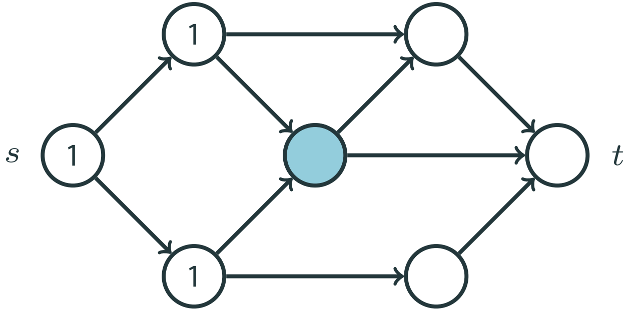

- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

# Число путей



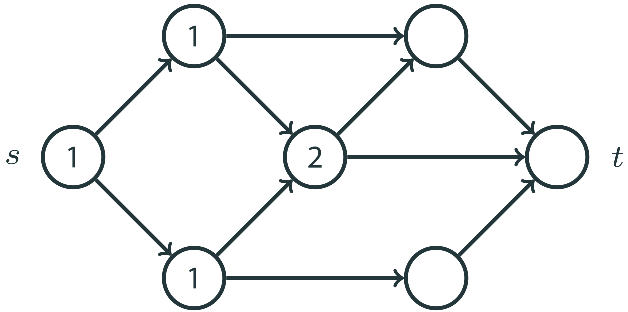

- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

# Число путей



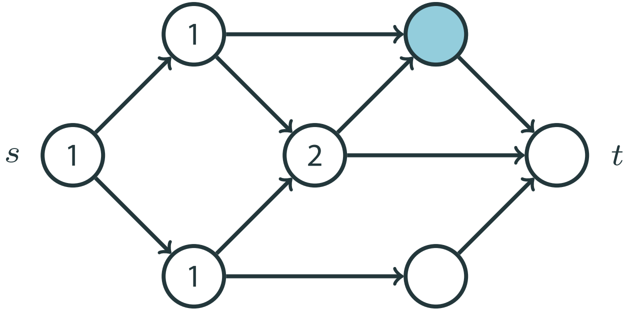

- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

# Число путей



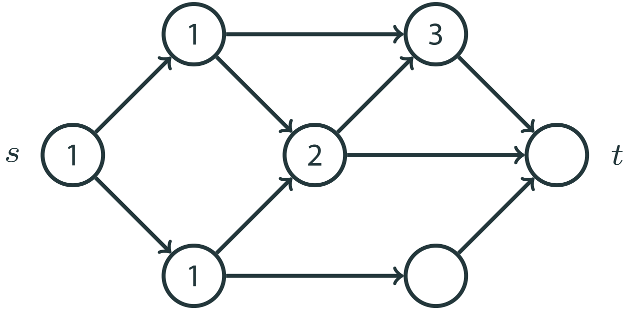

- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

# Число путей



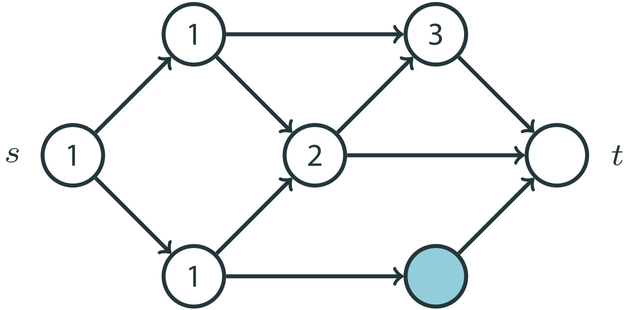

- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

# Число путей



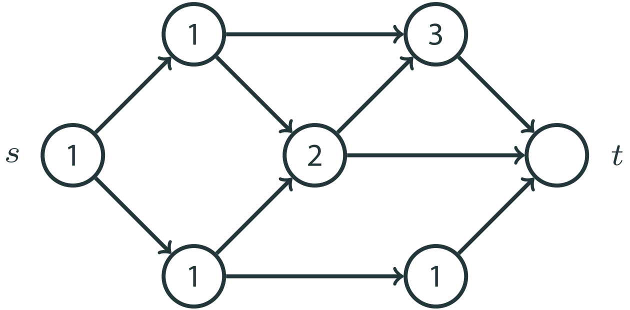

- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

# Число путей



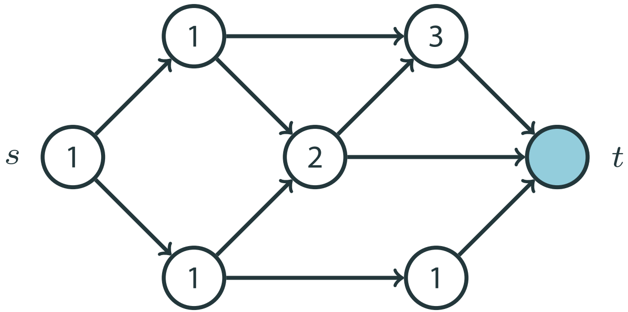

- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

# Число путей



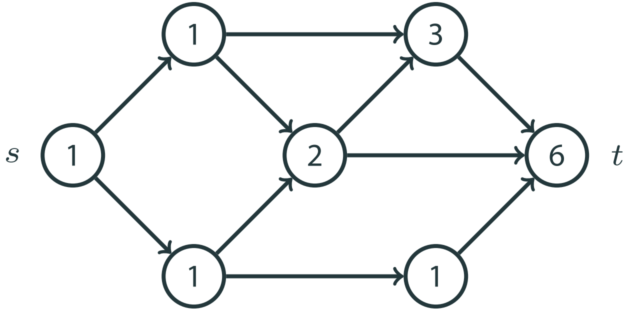

- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

# Число путей




- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

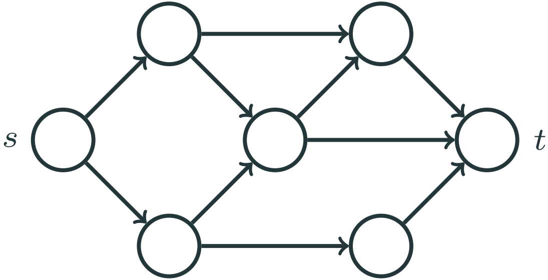
# Число путей




- Есть несколько разных путей; как посчитать их и ничего не пропустить?
- Мы можем посчитать их рекурсивно: для всякой точки посчитаем число путей из $s$ в данную точку
- Мы пользуемся правилом суммы!

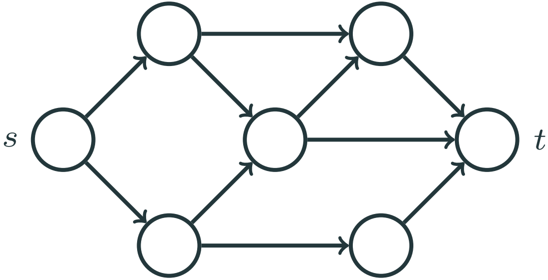
# Рекурсивные подсчеты

- Как могут возникать подобные задачи?



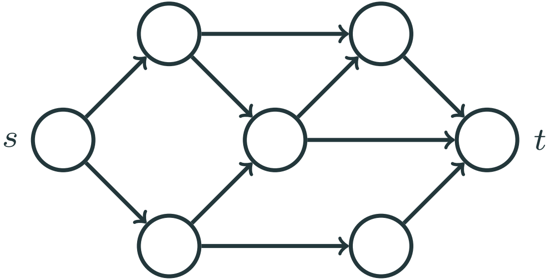
# Рекурсивные подсчеты

- Как могут возникать подобные задачи?
- Пусть у нас есть программа и вершины — ее блоки



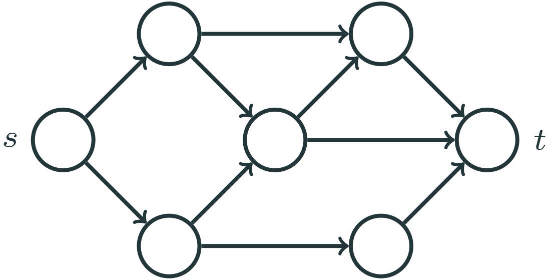
# Рекурсивные подсчеты

- Как могут возникать подобные задачи?
- Пусть у нас есть программа и вершины — ее блоки
- Запуск каждого блока запускает также все блоки, в которые из вершины ведет стрелка



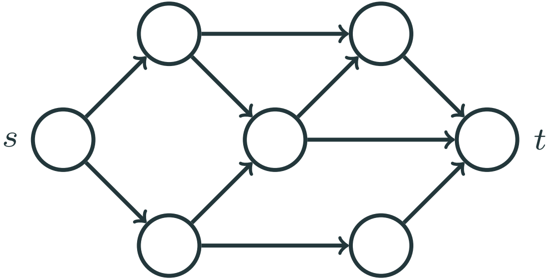
# Рекурсивные подсчеты

- Пусть мы запустили блок в вершине $s$



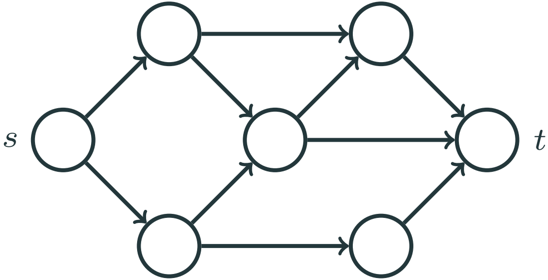
# Рекурсивные подсчеты

- Пусть мы запустили блок в вершине $s$
- Он запускает блоки в вершинах правее



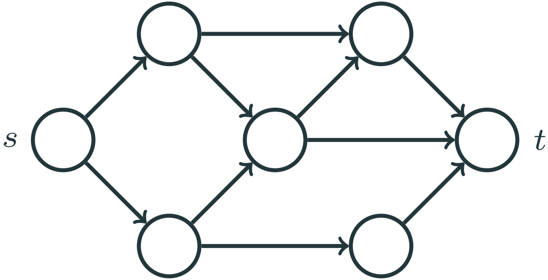
# Рекурсивные подсчеты

- Пусть мы запустили блок в вершине $s$
- Он запускает блоки в вершинах правее
- Те еще правее и так далее



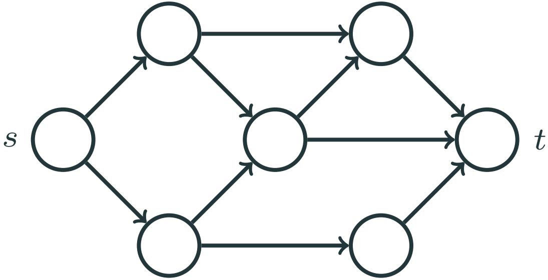
# Рекурсивные подсчеты

- Пусть мы запустили блок в вершине $s$
- Он запускает блоки в вершинах правее
- Те еще правее и так далее
- Сколько всего раз будет запущен блок $t$?



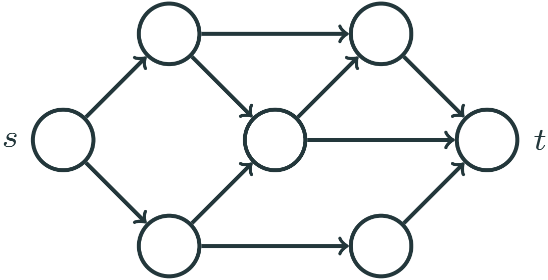
# Рекурсивные подсчеты

- Для каждого запуска блока $t$ можно проследить его происхождение



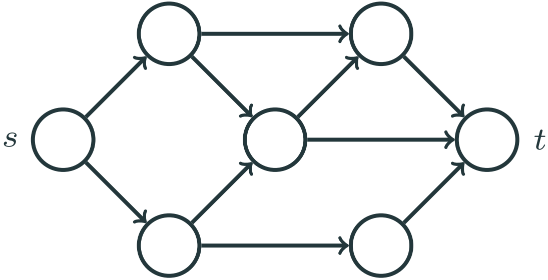
# Рекурсивные подсчеты

- Для каждого запуска блока $t$ можно проследить его происхождение
- Это путь из вершины $s$



# Рекурсивные подсчеты

- Для каждого запуска блока $t$ можно проследить его происхождение
- Это путь из вершины $s$
- Запусков столько, сколько путей



# Заключение

- Правило суммы очень простое

# Заключение

- Правило суммы очень простое
- Но даже его достаточно, чтобы делать нетривиальные подсчеты

# Заключение

- Правило суммы очень простое
- Но даже его достаточно, чтобы делать нетривиальные подсчеты
- Рекурсивные подсчеты могут быть очень полезны

# Заключение

- Правило суммы очень простое
- Но даже его достаточно, чтобы делать нетривиальные подсчеты
- Рекурсивные подсчеты могут быть очень полезны
- Мы увидим примеры позже в этом курсе

# Базовые подсчеты

- Зачем изучать подсчеты?
- Правило суммы
- Удобный язык: множества
- Обобщение правила суммы
- Рекурсивные подсчеты: число путей
- Правило произведения

# Правило произведения

**Правило произведения**

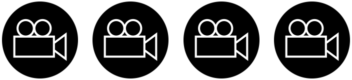
Если есть $k$ объектов первого типа и $n$ объектов второго типа, то есть $k \times n$ пар объектов, в которых первый объект первого типа, а второй — второго

# Правило произведения

## Правило произведения

Если есть $k$ объектов первого типа и $n$ объектов второго типа, то есть $k \times n$ пар объектов, в которых первый объект первого типа, а второй — второго

Видео



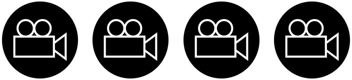
Пользователи

# Правило произведения

**Правило произведения**

Если есть $k$ объектов первого типа и $n$ объектов второго типа, то есть $k \times n$ пар объектов, в которых первый объект первого типа, а второй — второго

Видео



Пользователи

$4 \times 3 = 12$ оценок

# Список всех оценок

| | | | |
|---|---|---|---|
| 5 | 4 | 4 | 9 |
| 6 | 3 | 3 | 1 |
| 10 | 1 | 3 | 9 |

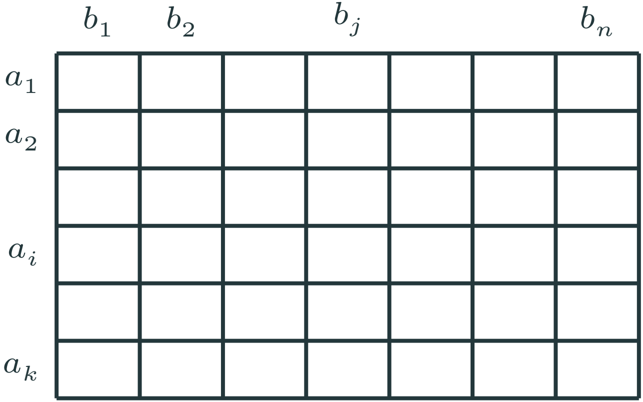
# Правило произведения на языке множеств

**Правило произведения**

Если множества $A$ и $B$ конечны, то есть $|A| \times |B|$ пар объектов, в которых первый объект из $A$, а второй из $B$

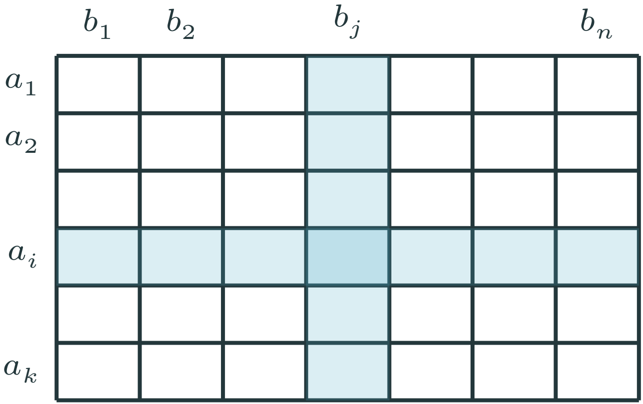
# Почему правило произведения верно?

$$A = \{a_1, \dots, a_k\}$$
$$B = \{b_1, \dots, b_n\}$$



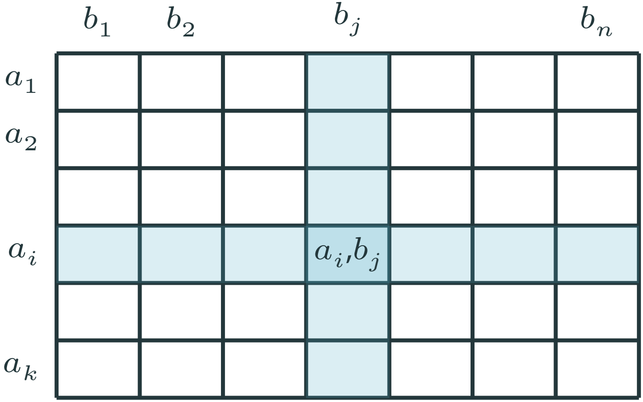
# Почему правило произведения верно?

$$A = \{a_1, \ldots, a_k\}$$
$$B = \{b_1, \ldots, b_n\}$$



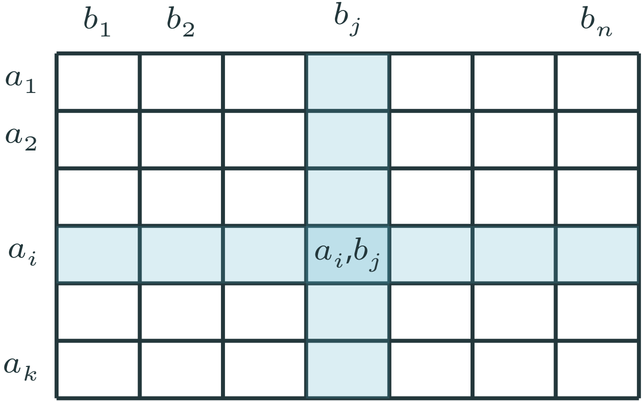
# Почему правило произведения верно?

$$A = \{a_1, \ldots, a_k\}$$
$$B = \{b_1, \ldots, b_n\}$$



# Почему правило произведения верно?

$$A = \{a_1, \dots, a_k\}$$
$$B = \{b_1, \dots, b_n\}$$




Пар столько же, сколько клеток в таблице

# Правило произведения как число путей

**Правило произведения**

Если множества $A$ и $B$ конечны, то есть $|A| \times |B|$ пар объектов, в которых первый объект из $A$, а второй из $B$

# Правило произведения как число путей

**Правило произведения**

Если множества $A$ и $B$ конечны, то есть $|A| \times |B|$ пар объектов, в которых первый объект из $A$, а второй из $B$

Можно ли представить правило произведения как подсчет числа путей?

# Правило произведения как число путей

**Правило произведения**

Если множества $A$ и $B$ конечны, то есть $|A| \times |B|$ пар объектов, в которых первый объект из $A$, а второй из $B$

# Правило произведения как число путей

## Правило произведения

Если множества $A$ и $B$ конечны, то есть $|A| \times |B|$ пар объектов, в которых первый объект из $A$, а второй из $B$

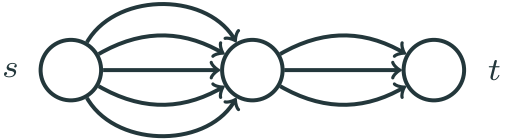
Для простоты рассмотрим $|A| = 5$ и $|B| = 3$

# Правило произведения как число путей

## Правило произведения

Если множества $A$ и $B$ конечны, то есть $|A| \times |B|$ пар объектов, в которых первый объект из $A$, а второй из $B$

Для простоты рассмотрим $|A| = 5$ и $|B| = 3$

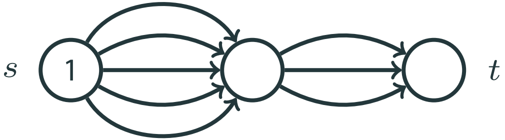


# Правило произведения как число путей

## Правило произведения

Если множества $A$ и $B$ конечны, то есть $|A| \times |B|$ пар объектов, в которых первый объект из $A$, а второй из $B$

Для простоты рассмотрим $|A| = 5$ и $|B| = 3$

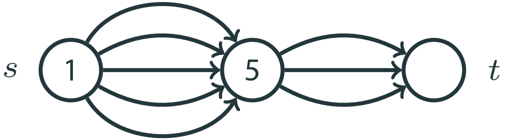


# Правило произведения как число путей

**Правило произведения**

Если множества $A$ и $B$ конечны, то есть $|A| \times |B|$ пар объектов, в которых первый объект из $A$, а второй из $B$

Для простоты рассмотрим $|A| = 5$ и $|B| = 3$



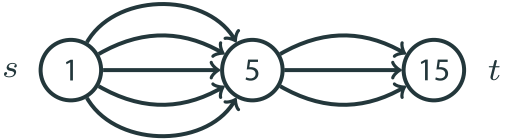

$$1 + 1 + 1 + 1 + 1 = 5$$

# Правило произведения как число путей

## Правило произведения

Если множества $A$ и $B$ конечны, то есть $|A| \times |B|$ пар объектов, в которых первый объект из $A$, а второй из $B$

Для простоты рассмотрим $|A| = 5$ и $|B| = 3$




$$5 + 5 + 5 = 3 \times 5 = 15$$

# Заключение

- Подсчеты начинаются с простых наблюдений

# Заключение

- Подсчеты начинаются с простых наблюдений
- Но даже в правиле суммы есть тонкости

# Заключение

- Подсчеты начинаются с простых наблюдений
- Но даже в правиле суммы есть тонкости
- Уже очень простые идеи могут быть полезны

# Заключение

- Подсчеты начинаются с простых наблюдений
- Но даже в правиле суммы есть тонкости
- Уже очень простые идеи могут быть полезны
- Дальше мы увидим, как более сложные конструкции строятся на тех базовых идеях, которые мы обсудили