Основы теории принятия решений

В.О. Гроппен

Лекция 1

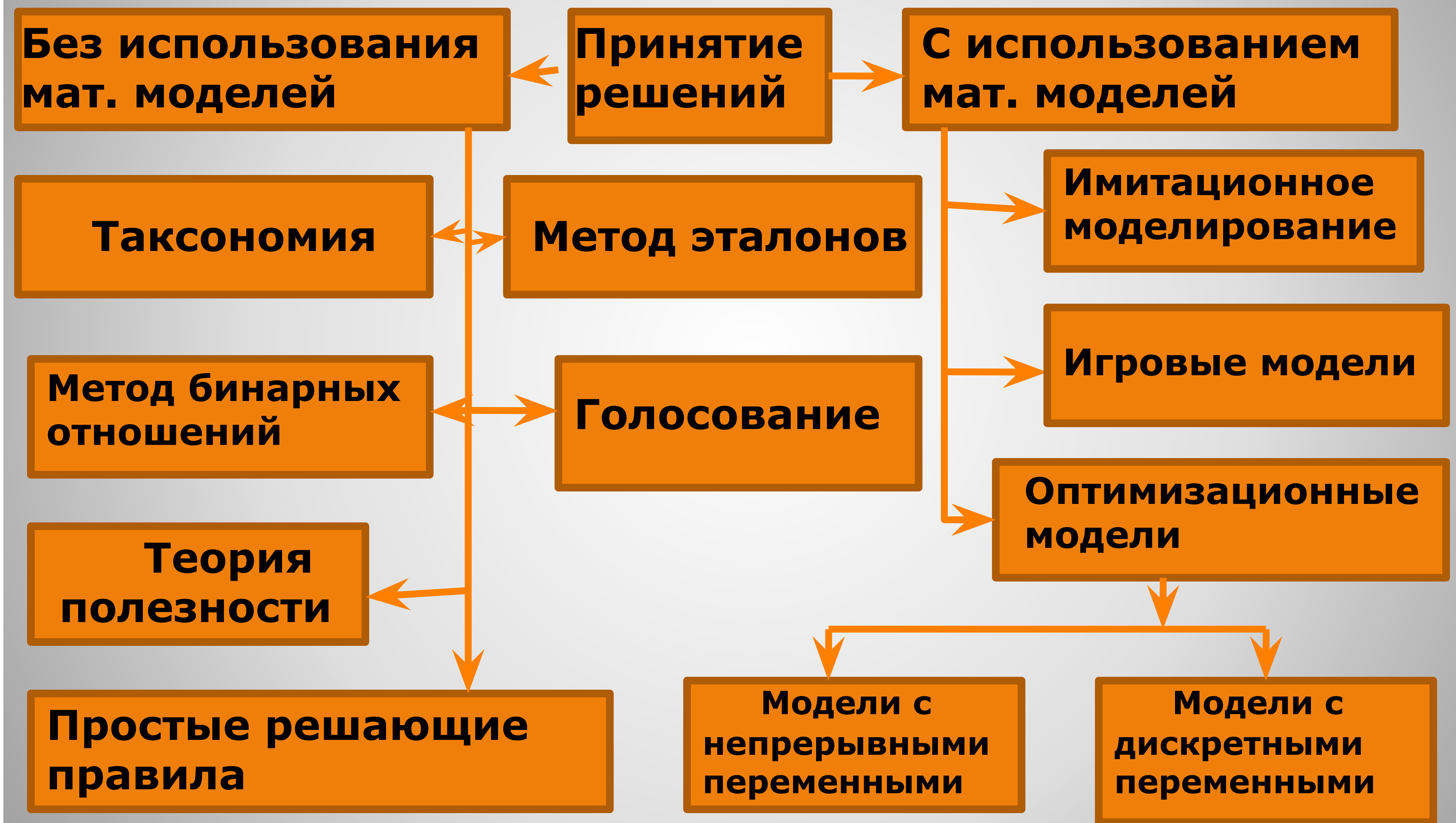
Часть 1

Обзор тем курса ТПР

Необходимое условие принятия решений

- **Необходимое условие принятия решений – наличие альтернатив выбора**

Классификация технологий принятия решений



Простые решающие правила

1. **Уставы** (воинские уставы, устав СКГМИ и других университетов, монастырские уставы…). Привести примеры самостоятельно.
2. **Кодексы** (гражданский кодекс, кодекс чести…) Привести примеры самостоятельно.
3. **Правила поведения**: в общежитии, в

Древнем Риме:

Лучше быть первым на селе, чем вторым в Риме.

Родина там, где хорошо.

Предупрежден – значит вооружен.

Привести примеры самостоятельно

Таксономия

Типы задач таксономии:

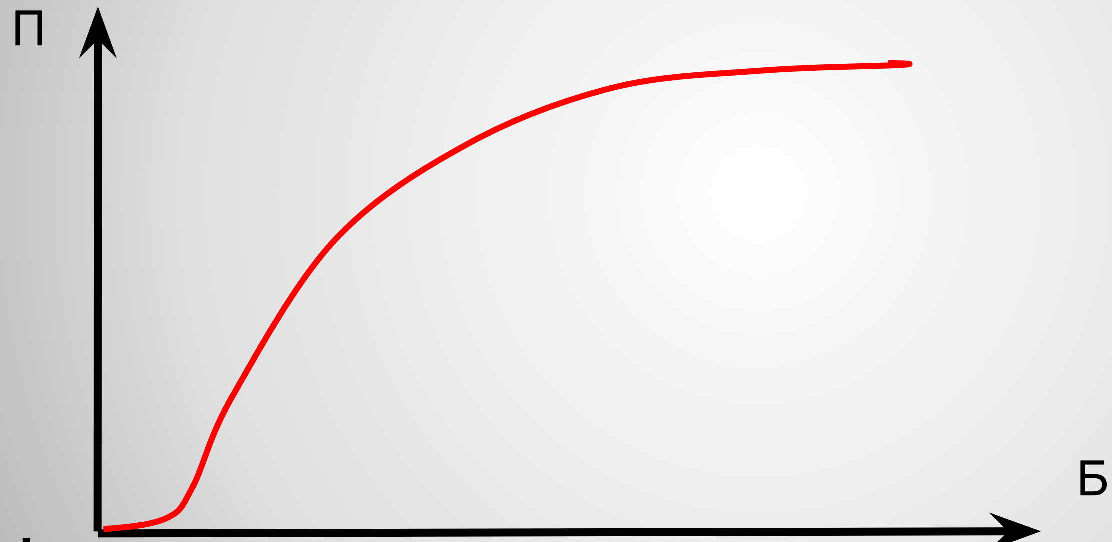
- Объединение статичных объектов в таксоны по «похожести».
- Объединение статичных объектов в заданное число таксонов.
- Объединение динамичных объектов в таксоны.
- Выделение устойчивых таксонов.
- Ранжирование объектов.
- Прогнозирование свойств объектов.

Метод бинарных отношений

- Основная задача метода бинарных отношений – ранжирование объектов на основании качественных данных о парных предпочтениях.

Использование теории полезности

- Полезность богатства:



- Цель: дать количественную оценку отношениям предпочтения.

Принятие решений голосованием

Решаемые задачи:

1. **Способы организации голосования.**
2. **Способы подведения итогов голосования**
3. **Технология прогнозирования итогов голосования.**

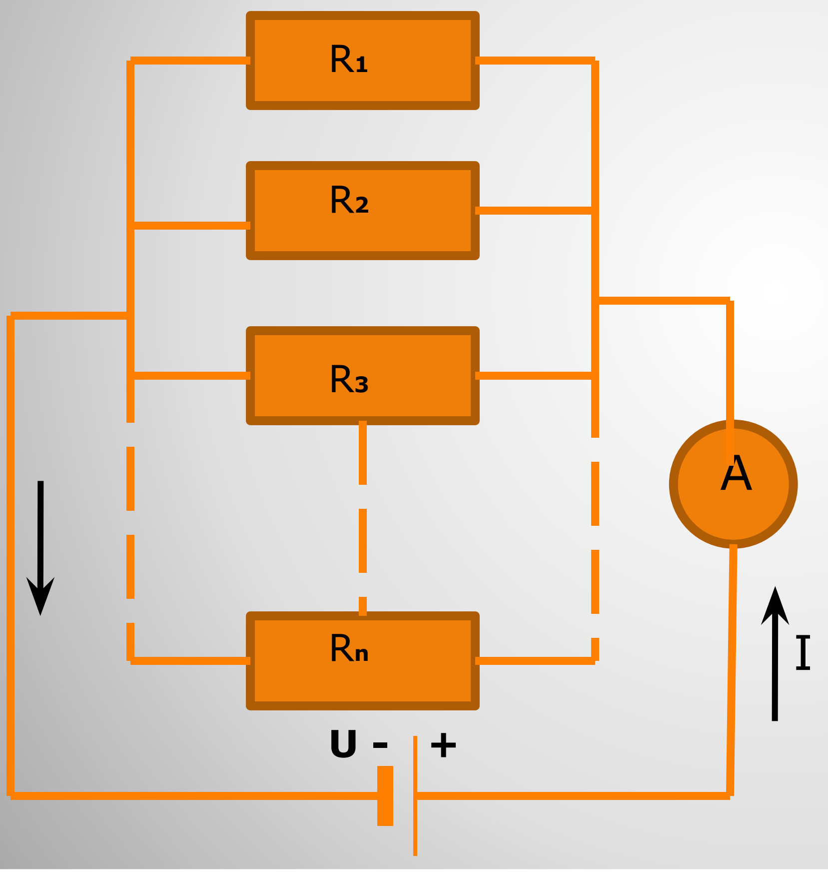
Метод эталонов

- **Решаемые задачи:**

1. **Ранжирование многокритериальных объектов.**
2. **Обработка экспертных оценок.**
3. **Подведение итогов голосования.**
4. **Прогнозирование персональной успеваемости учащихся.**
5. **Выбор направлений развития науки и технологий.**

Имитационное моделирование

Электрическая схема



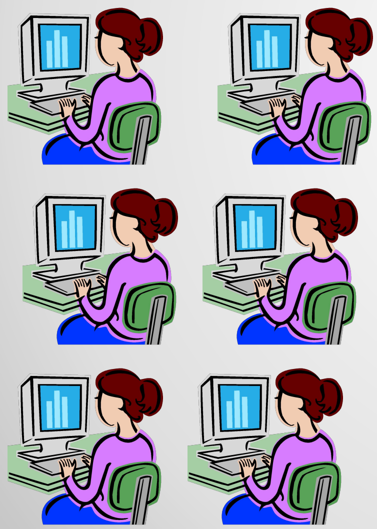
Математическая модель

$$\begin{cases} I = \dfrac{U}{R}; \\ \dfrac{1}{R} = \displaystyle\sum_{j=1}^{j=n} \dfrac{1}{R_j}. \end{cases}$$

Игровое моделирование

Выбор метода обучения преподавателем

Матричная антагонистическая игра двух лиц

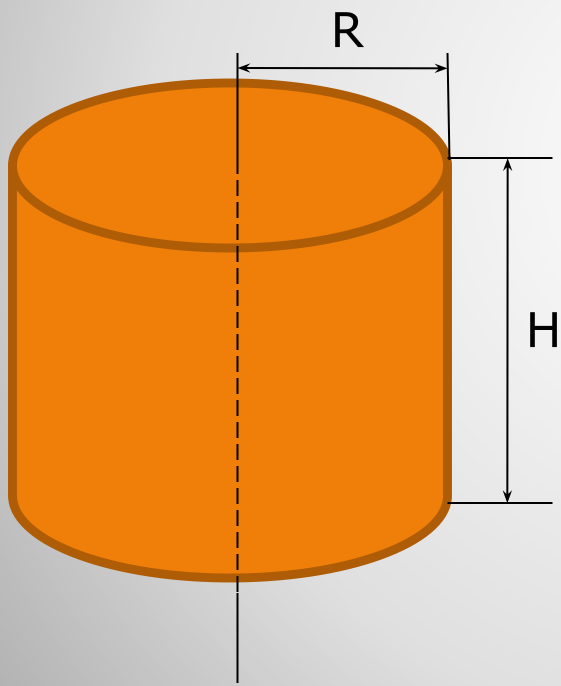


$$T_{opt} = \min_{j} \max_{i} t_{i,j}$$

t_{11}	t_{12}	t_{13}
t_{21}	t_{22}	t_{23}
t_{31}	t_{32}	t_{33}
t_{41}	t_{42}	t_{43}
t_{51}	t_{52}	t_{53}
t_{61}	t_{62}	t_{63}

Оптимизационные задачи с непрерывно меняющимися переменными

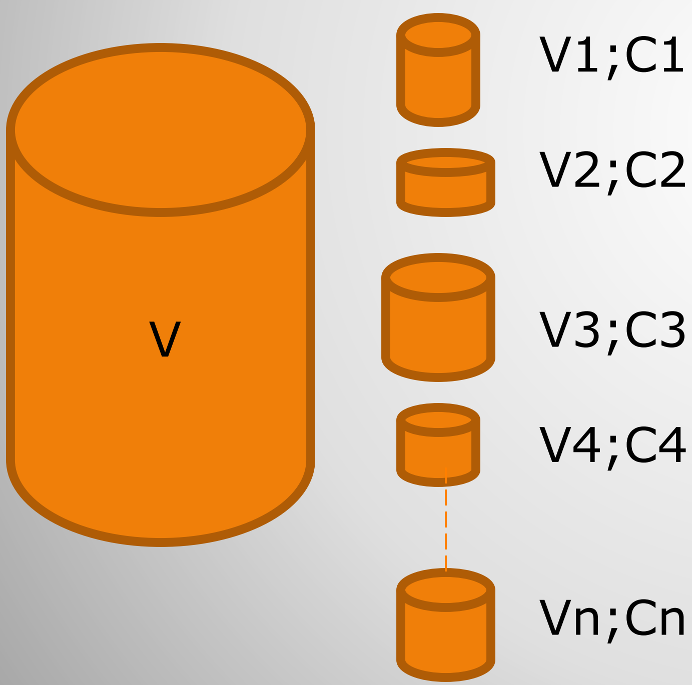
Задача о консервной банке



$$\begin{cases} S = 2\pi(R^2 + RH) \to \min; \\ \pi R^2 H = V; \\ R \geq 0; H \geq 0. \end{cases}$$

Оптимизационные задачи с дискретно меняющимися переменными

Задача о ранце



$$\begin{cases} \sum\limits_i C_i z_i \to \max; \\ \sum\limits_i V_i z_i \le V; \\ \forall i : z_i = 1,0. \end{cases}$$

Повторить самостоятельно курсы:

- Теория графов
- Теория множеств
- Математическая логика
- Методы оптимизации
- Мат. анализ

Часть 2

Простые алгоритмы таксономии

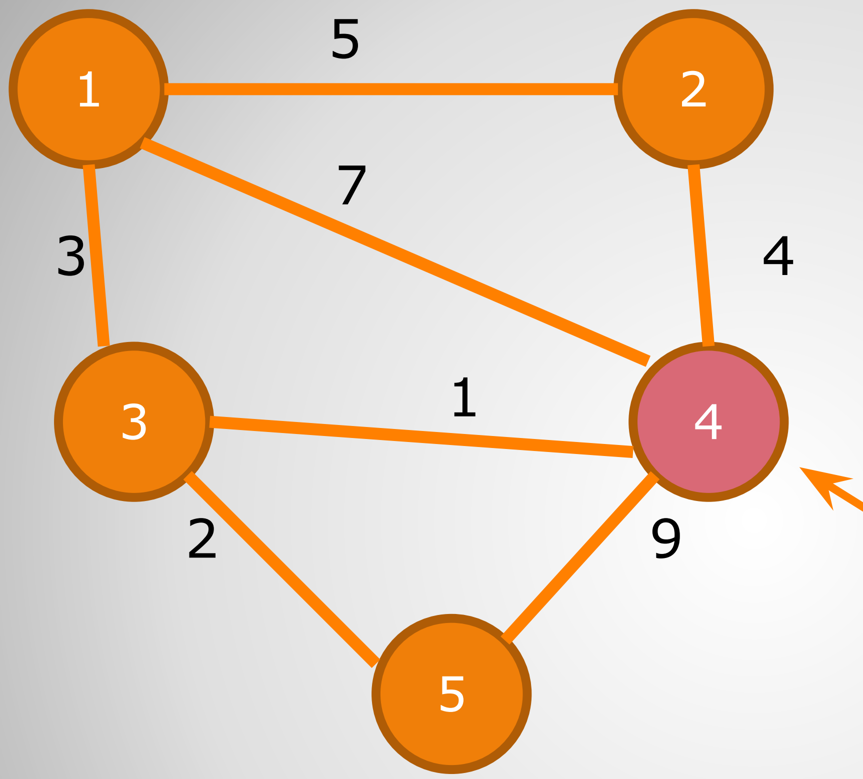
Алгоритм Прима

1. На взвешенном неориентированном графе выбирается произвольная вершина.
2. Выбирается вершина, расстояние до которой от исходной вершины минимально (т.е. вес ребра минимален).
3. Выделяется ребро, соединяющее две выбранные выше вершины.
4. Для выделенного ребра:

   1. Добавляем вес этого ребра к ранее накопленной сумме.

   2. «Стягиваем» вершины, принадлежащие выбранному ребру, в одну вершину.

   3. Если образуются параллельные ребра, то остается лишь то из них, вес которого минимален, а остальные удаляются.
5. Если весь граф стянут в одну вершину, то перейти к шагу 6, нет – к шагу 1.
6. «Стянутые» ребра представляют собой минимальный остов графа.
7. Конец алгоритма.

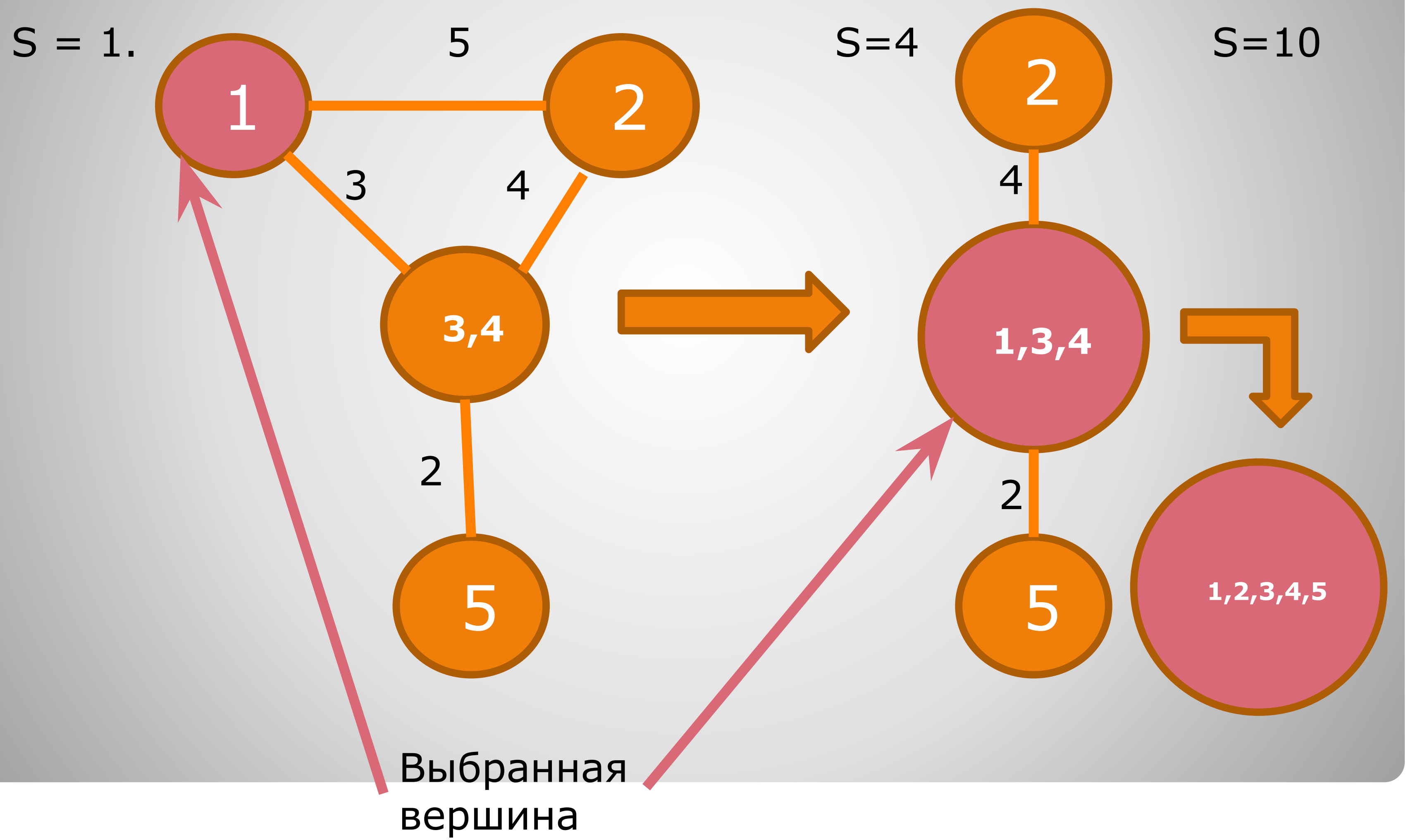


Исходный граф G(X,U)

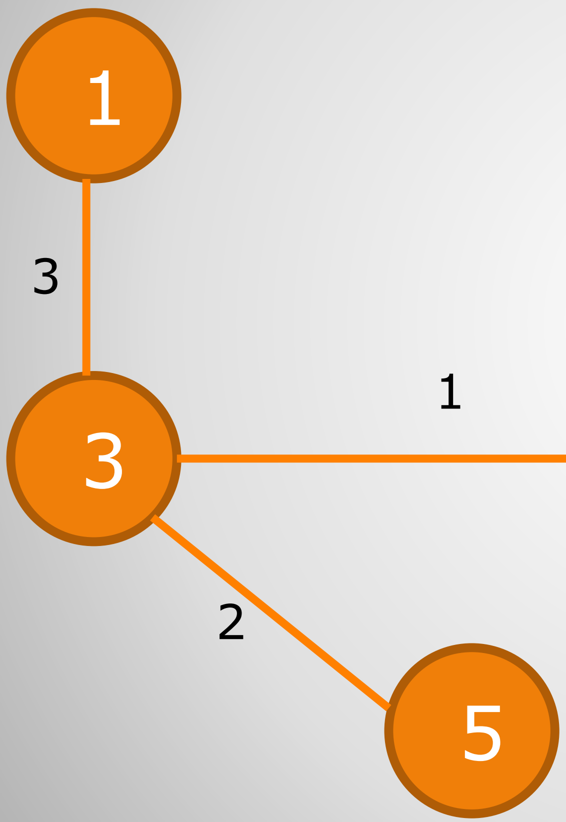
Стартовая вершина

Пример работы алгоритма Прима

Пример работы алгоритма Прима 2

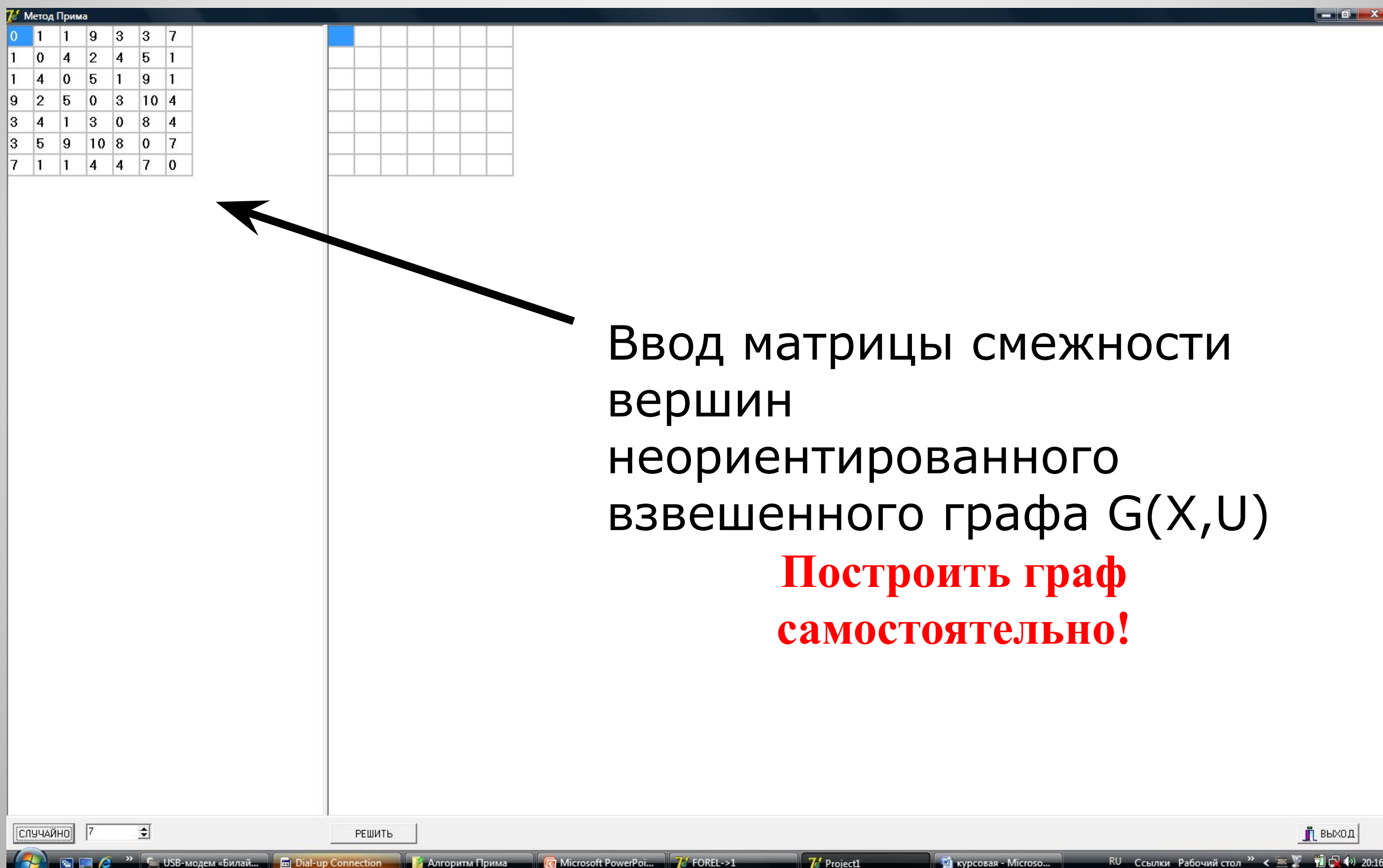


Минимальный остов графа G(X,U)



S=10

Интерфейс Программной реализации алгоритма Прима 1

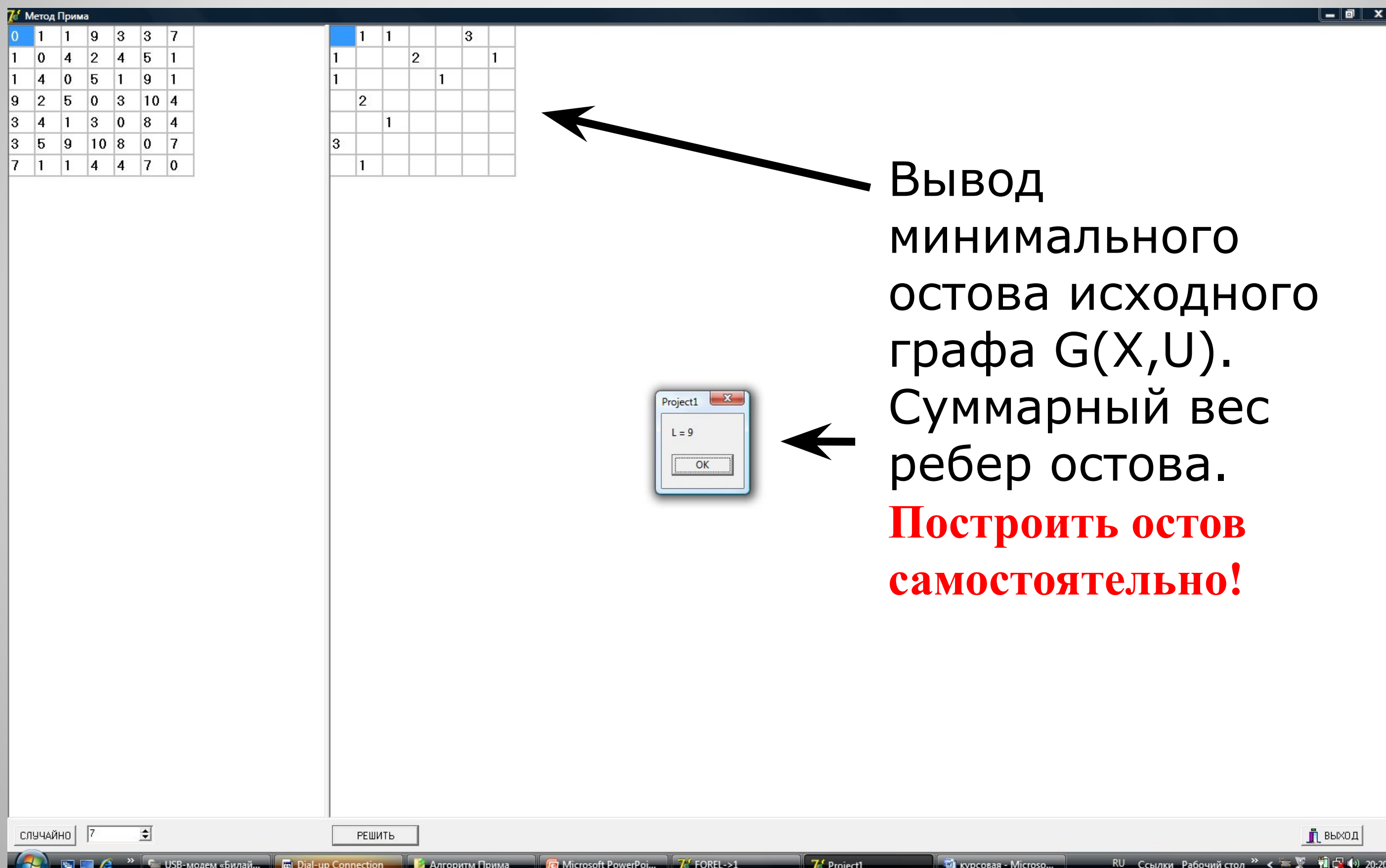


0	1	1	9	3	3	7
1	0	4	2	4	5	1
1	4	0	5	1	9	1
9	2	5	0	3	10	4
3	4	1	3	0	8	4
3	5	9	10	8	0	7
7	1	1	4	4	7	0

Ввод матрицы смежности вершин неориентированного взвешенного графа G(X,U)

Построить граф самостоятельно!

Интерфейс Программной реализации алгоритма Прима 2

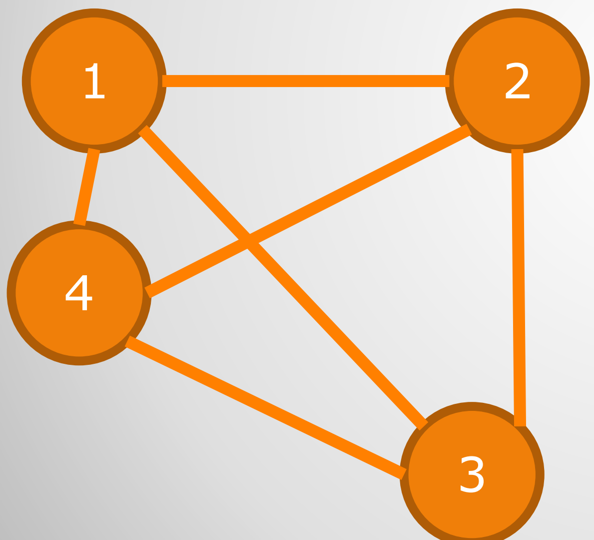


Вывод минимального остова исходного графа G(X,U). Суммарный вес ребер остова.

Построить остов самостоятельно!

Алгоритмы таксономии

- Пусть заданы: множество точек (вершин) A и множество ребер (i,j), где r(i , j) - длина ребра (i , j).



D – длина самого длинного ребра.

Нормализация:

$d(i,j)=r(i,j)/D$

Определение λ - расстояний

- Шаг 1.Выбирается любая, ранее не просматривавшаяся пара точек p и q. Если таковых нет, то перейти к шагу 5.
- Шаг 2.Среди рёбер, смежных (p,q) выбирается самое короткое, длину которую обозначаем B.
- Шаг 3. $\lambda(i,j)=d(i,j)/B$.
- Шаг 4. Перейти к шагу 1.
- Шаг 5. Конец алгоритма.

Самостоятельно

Определить λ – расстояние между 3-й и 5-й вершинами на графе, заданном матрицей М:

0	2	7	5	9	10
2	0	3	8	1	6
7	3	0	4	11	2
5	8	4	0	8	5
9	1	11	8	0	17
10	6	2	5	17	0

Гипотеза λ - компактности

- Гипотеза λ - компактности формулируется следующим образом: реализация одного и того же образа обычно отражается в признаковом λ - пространстве в «близких» точках, образуя λ - компактные сгустки.
- Для определения λ - расстояния в λ - пространстве используется алгоритм Прима

САМОСТОЯТЕЛЬНО

- Объединить в таксоны в λ - пространстве трёх учеников - отличника, хорошиста и двоечника (по две оценки у каждого), если в один таксон объединяются лица, λ - расстояния между которыми меньше 0.1.

Назначение и свойства алгоритма Forel 1

- 1. Алгоритм Forel 1 предназначен для разбиения объектов на таксоны.
- 2. Форма всех таксонов – сфера (гиперсфера).
- 3. Радиусы таксонов известны.
- 4. Число таксонов a priori неизвестно.

Forel-1 (шаги 1 – 5)

- Шаг 1. Все признаки объектов нормируются так, чтобы их значения были в диапазоне 0-1.
- Шаг 2. R0=+∞.
- Шаг 3. Все точки считаем непомеченными.
- Шаг 4. На множестве непомеченных точек выбирается произвольная **x**i, после чего осуществляется переход к шагу 5. Если таковых точек нет, то перейти к шагу 8.
- Шаг 5. Ищется максимальное расстояние R от **xi** до остальных точек.

Forel-1 (шаги 6 – 13)

- Шаг 6. R0= min{R0; R}.
- Шаг 7. Точка **x**i помечается. Если помечены все точки, то перейти к шагу **8**, нет - к шагу **4.**
- Шаг 8. R=R0 – ε.
- Шаг 9. Если множество точек пусто. То перейти к шагу **16,** нет - к шагу **10**.
- Шаг 10. Все точки считаем непомеченными.
- Шаг 11. На множестве непомеченных точек выбирается произвольная точка **xi**.
- Шаг 12. Определяется число точек, расстояние которых до не превышает R.
- Шаг 13. Точку считаем помеченной. Если помечены все точки, то перейти к шагу **14**, нет -к шагу **11**.

Forel-1 (шаги 14 – 16)

Шаг 14. Выбор j-й точки, для которой величина P(j) минимальна.
Шаг 15. Все точки, расстояние до которых от j-й точки не превышает R, удаляются. Перейти к шагу 9.
Шаг 16. Конец алгоритма.

Достоинства и недостатки алгоритма Forel 1

- **Достоинства:**

1. Простота.
2. Легкость программной реализации.

- **Недостатки:**

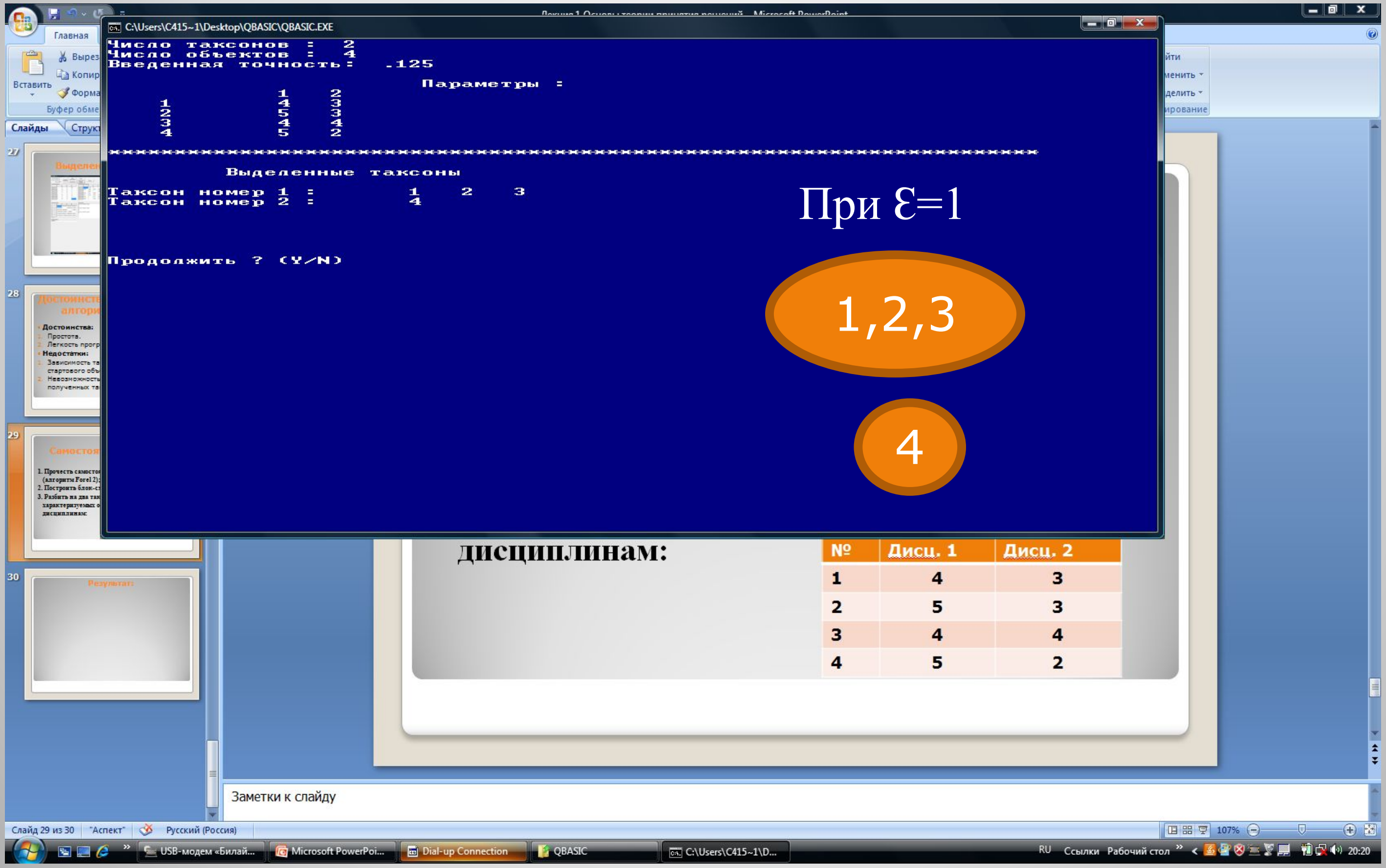
1. Зависимость таксономии от выбора стартового объекта.
2. Невозможность контролировать число полученных таксонов.

Самостоятельная работа

Разбить на таксоны группы из четырех, следующих один за другим учеников, характеризуемых оценками по трем дисциплинам. В один таксон включаются ученики, «расстояние» между которыми не превышает двух.

№	Дисц. 1	Дисц. 2	Дисц. 3	№	Дисц. 1	Дисц. 2	Дисц. 3
1	2	3	4	12	4	2	5
2	2	3	5	13	4	3	2
3	2	4	3	14	4	3	5
4	2	4	5	15	4	5	3
5	3	2	4	16	4	5	2
6	3	2	5	17	5	2	3
7	3	4	2	18	5	2	4
8	3	4	5	19	5	3	2
9	3	5	4	20	5	3	4
10	3	5	2	21	5	4	3
11	4	2	3	22	5	4	2

Результат решения программой Forel 2:



Самостоятельно:

- Программно реализовать алгоритм Forel 1.
- Исследовать программу и построить графические зависимости времени работы программы от размерности решаемой задачи.