“运筹选律，智启未来”

清华大学 Tsinghua University

多目标个性化课程规划系统

计算思维小组专题报告

小组成员：赵乐毅 刘宪武 卢绪涵

2025年12月30日

目录 CONTENTS

01

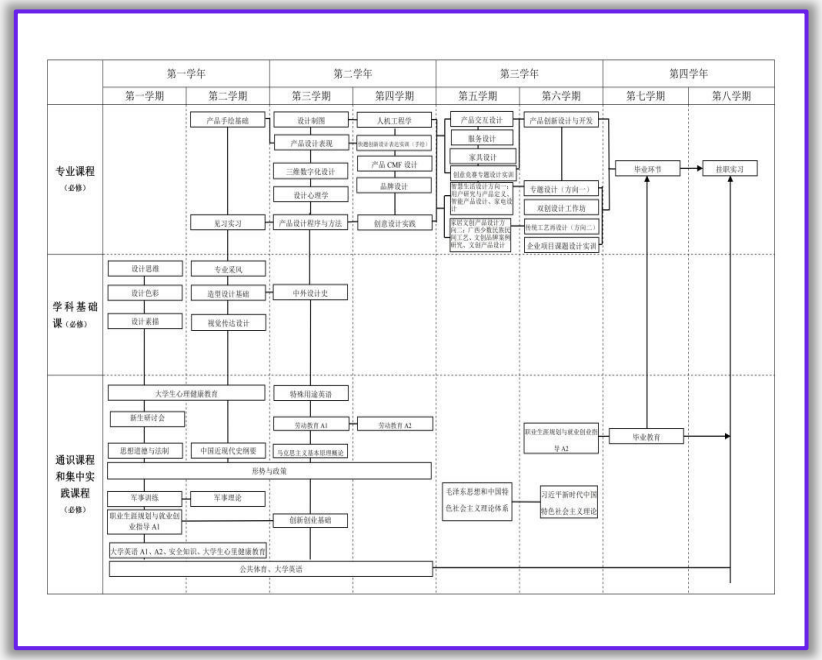
问题切入

探索之旅，从选课开始

谈到选课...

“限选到底是什么呀？”
“我全都想选！”
“完了挂科了，补修怎么安排？”

- 符合**培养方案**，遵循先修关系
- **个人精力**有限，并非多多益善
- 处理**意外**发生，动态调整方案



让我们先解决“选课”这个问题！

从直觉到数据

“应该能承受这些课吧…”

“我赌我能抢到这门课！”

“我真的需要这门课吗？”

直觉、经验

- 清晰的目标定义（我要什么？）
- 量化的约束识别（我能承受什么？）
- 系统的方案生成（怎样最优？）
- 动态的调整能力（变了怎么办？）

数据、组合、计算

穿针引线，利用好我们已有的信息！

02

解决思路

从直觉到计算

由浅入深三步走

建图建模

- 图结构
- Kahn算法

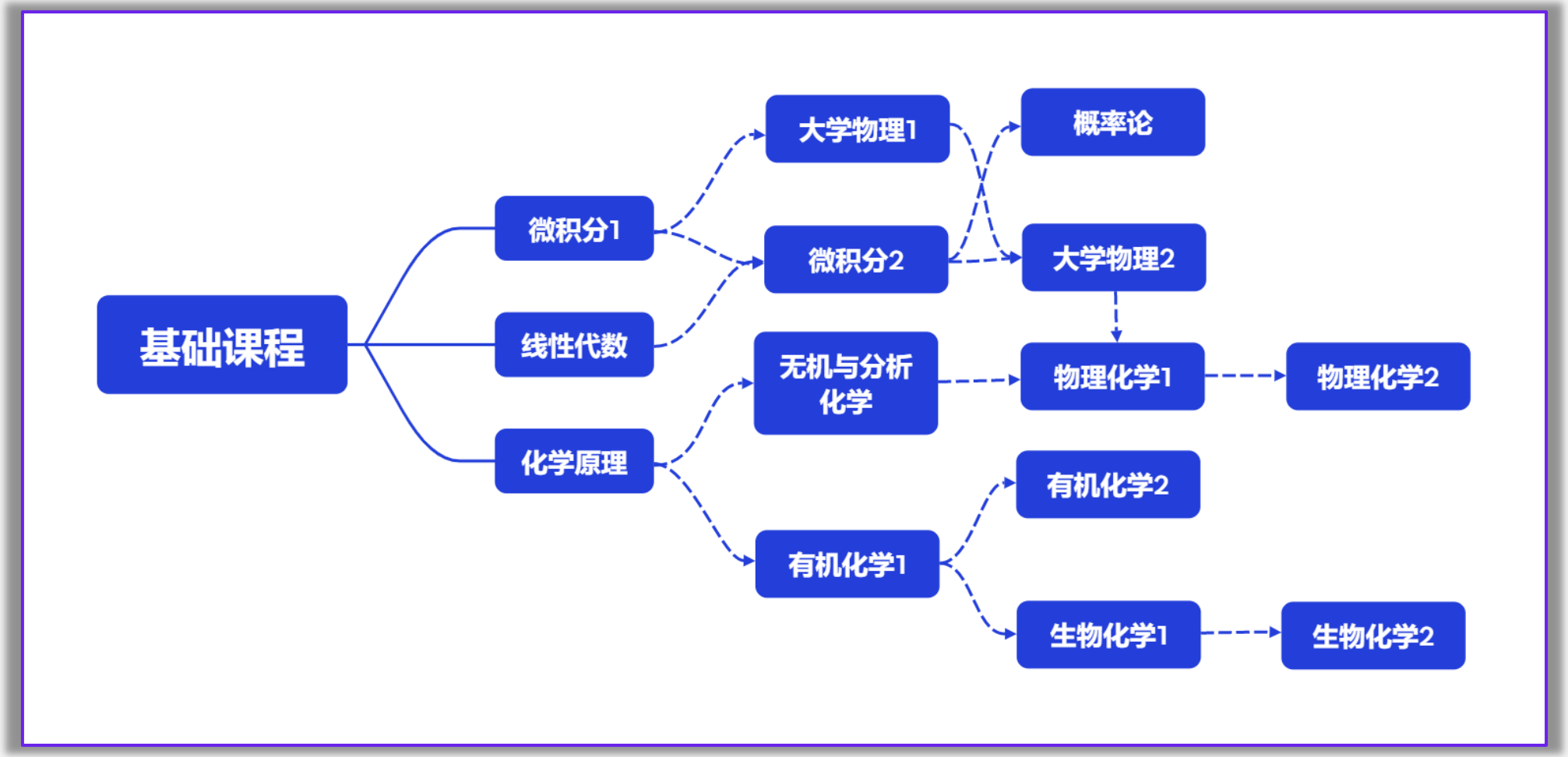
量化目标

- 学业负荷
- 难度衡量

个性适配

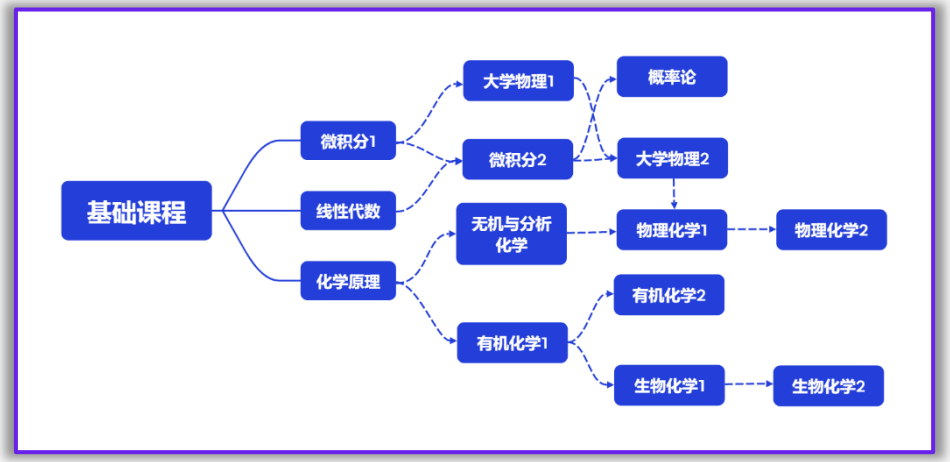
- 专业需求
- 兴趣匹配

如何得到一个符合逻辑的课程顺序?



如何得到一个符合逻辑的课程顺序?

- 每个课程都是一个节点
- 先修关系相当于有向边
- 整体形成有向无环图（DAG）

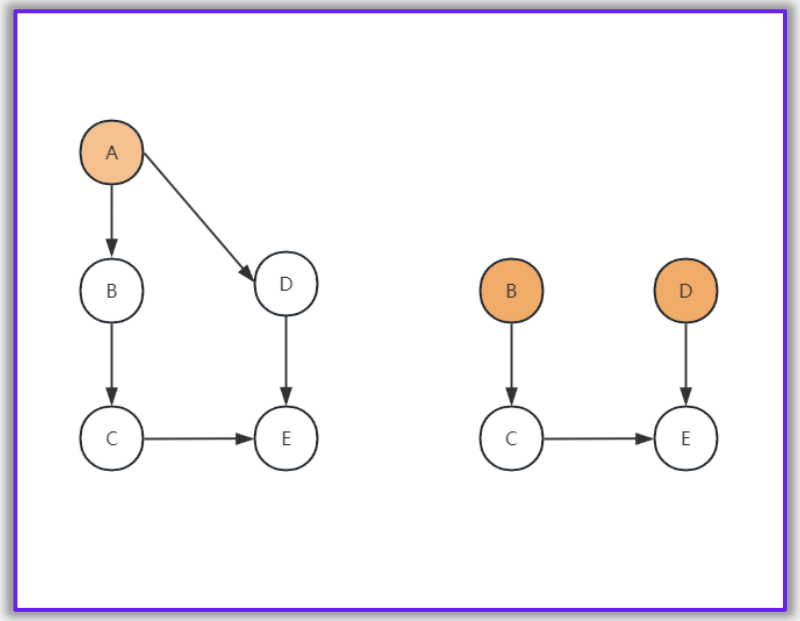


一个直观的想法是，先找到“我能学的课”，再进一步看“学了这个我就能学什么课”。这样的想法该如何实现?

Kahn 算法

在一个有向图中，一个顶点的出度指的是由该顶点指出的边的总数；一个顶点的入度为指向该顶点的边的总数。

1. 统计每个顶点的入度，统计每个顶点指出的边的终点集合
2. 遍历顶点的入度统计结果，将入度为0的顶点全部加入队列
3. 遍历从队列头部移出的顶点
4. 对于移出队列头部的顶点而言，先将其追加到排序结果当中；
   然后，对所有以该顶点为起点的边的终点而言，将其入度均减1。
   如果入度减为0，则将相应顶点加入队列当中
5. 重复步骤3～4，直到队列为空为止
6. 判定排序结果中顶点的数量是否等于有向图的顶点总数



https://zhuanlan.zhihu.com/p/644881066

Kahn 算法

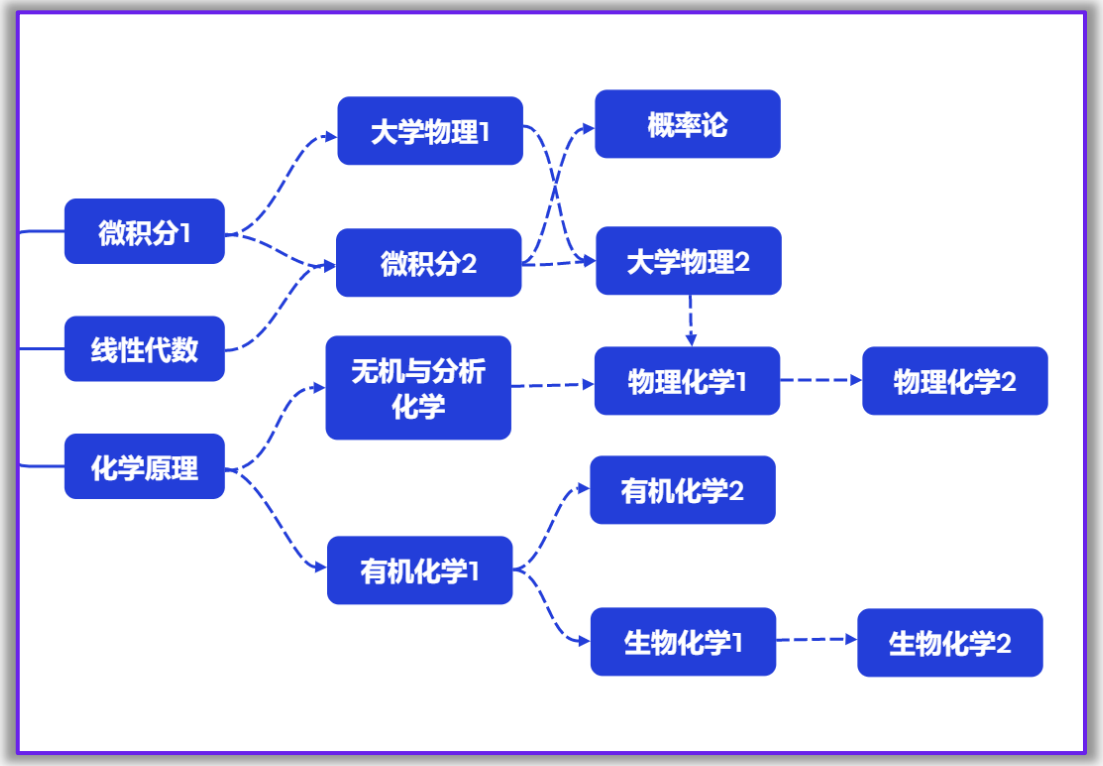
Algorithm 1: Kahn's Algorithm: Topological Sorting

Input : Directed acyclic graph $G = (V, E)$, where V is the vertex set and E is the edge set

Output: Topological ordering L, or detection that the graph contains a cycle

```
Compute the in-degree in-degree[v] for each vertex v ∈ V
Initialize a queue Q (to store all vertices with in-degree 0)
Initialize an empty list L (to store the topological ordering)
foreach vertex v ∈ V do
    if in-degree[v] = 0 then
        Q.push(v)
while Q is not empty do
    u ← Q.pop() L.append(u)
    foreach edge (u, v) ∈ E (for each adjacent vertex v of u) do
        in-degree[v] ← in-degree[v] − 1 ;          // Remove edge (u, v)
        if in-degree[v] = 0 then
            Q.push(v)
if |L| ≠ |V| then
    return "Graph contains a cycle, topological sorting not possible"
else
    return L
```

Kahn 算法



- 利用Kahn算法，我们可以得到以下顺序：
- 微积分1、线性代数、化学原理、大学物理1...

为了得到多种可能序列，可以用**随机数**或者**优先级**...

（较为合理的）量化指标

一些很有用的量化指标：基础属性、实践性、负荷指数、给分水平、难度[1][2]、兴趣匹配度。

这些指标有些容易获取，有些不易获取，还有一些依赖主观估计...

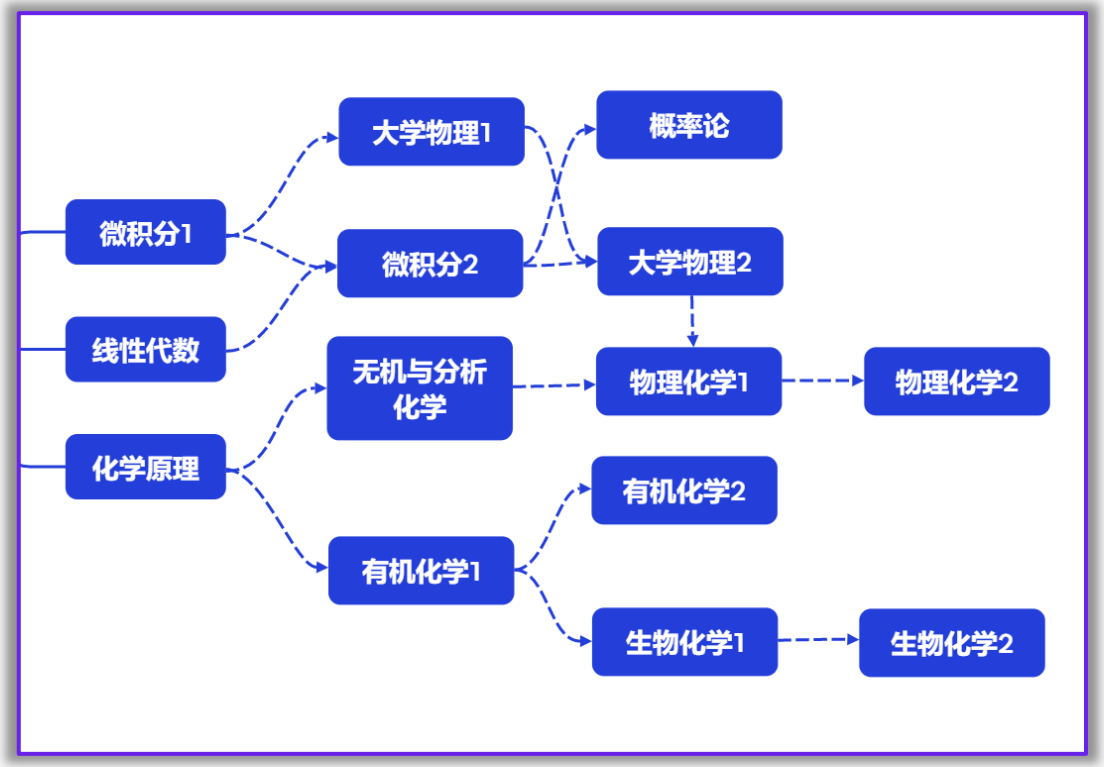
$$\text{负荷指数} = \frac{\text{课外学时数}}{\text{课内学时数}}$$

$$\text{学业压力} = \sum \text{负荷指数} \times \text{学分}^2$$

口 先保证我们的估计是合理的！

[1]施美玲,杨桃香.基于成对比较的课程难度评估[J].曲靖师范学院学报,2017,36(06):49-54.
[2]俞晓明,曹玉娟,谢丽,等.课程难度研究综述[J].物理教师,2017,38(05):2-7+12.

优化与抉择



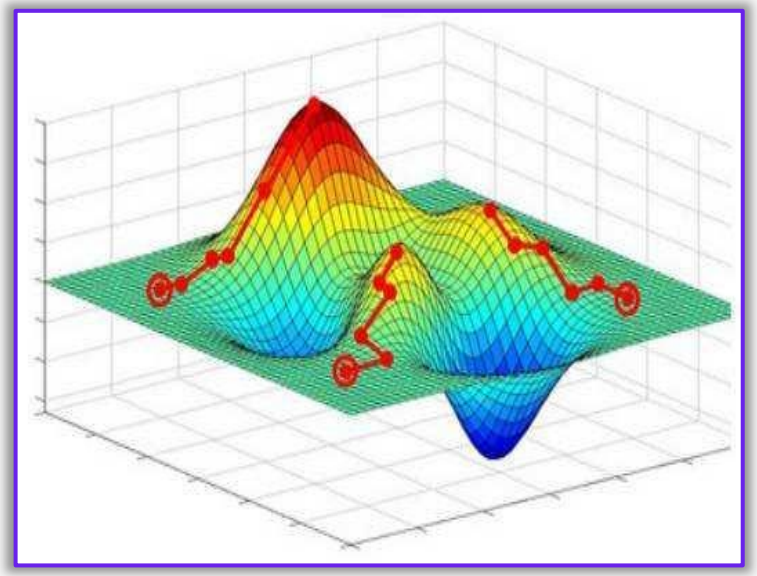

```
Active code page: 65001
输入课程数n: 14
输入每门课的学分（共14个）: 5 4 4 4 5 2 3 5 4 4 3 3 3 3
输入每学期学分上限M: 15
输入学期数上限K: 5
输入先修关系数量: 14
输入先修关系（每行两个整数a b，表示a是b的先修课，课程编号从1开始）:
1 4
1 5
2 5
3 6
3 7
4 9
5 8
5 9
6 10
7 11
7 12
10 13
12 14
9 10
排课方案数: 70
```

优化与抉择

对于特定的目标，在众多选择中总存在最符合我们需求的那一个！

但是情况并不会这么简单，我们还会面临以下问题：

- 在浩如烟海的方案中，**如何快速找出**最符合我们需求的那一个？
- 如果我们很贪心，**既**有这个需求，**又**有那个需求，怎么办？



03

实践方法

随你而变的运筹选律

数据准备：如何结构化课程信息

课程基础信息

- 课程ID、名称、学分、开课学期、先修课程列表
- 负荷指数、给分水平、兴趣标签等量化字段

先修关系图

- 通过先修课程ID, 后续课程ID来存储
- 可用NetworkX库

算法实现：图遍历与优化算法

- 图如何遍历？Kahn算法
- 选取优化目标？明确目的，寻找合适的优化量
- 如何初步优化？在遍历过程中按照优先级排序！
- 优化算法如何选取？单目标、多目标...

Algorithm 1 基于多目标优先级的个性化排课算法 (Prioritized Course Scheduling)

Require: $G(V, E)$: 课程有向无环图，其中 V 为课程集合，E 为先修关系；C_{max}: 每个学期的最大允许学分上限；W: 用户偏好权重向量 (用于计算综合优先级分数)；S_{total}: 计划总学期数。

Ensure: $Plan$: 一个列表，包含每个学期的课程安排 $\{S_1, S_2, ..., S_{total}\}$。

Initialization:
计算所有课程的入度: $InDegree[v] \leftarrow \text{count\_parents}(v), \forall v \in V$
初始化候选课程集合 (优先队列): $Q \leftarrow \emptyset$ 将所有无先修课的课程加入队列:
for $v \in V$ **do**
  if $InDegree[v] == 0$ **then**
    Q.push(v)
  end if
end for
$Plan \leftarrow$ list of empty sets
Main Loop:
for $t \leftarrow 1$ to S_{total} **do**
  $CurrentSemesterLoad \leftarrow 0$
  $SelectedCourses \leftarrow \emptyset$ ▷ 1. 优化目标计算与排序
  for $v \in Q$ **do**
    $Score[v] \leftarrow \text{CalculatePriority}(v, W)$
  end for
  Sort Q based on $Score$ in descending order ▷ 2. 贪心选择当前学期课程
  for $v \in Q$ **do**
    if $CurrentSemesterLoad + v.credits \leq C_{max}$ **then**
      $SelectedCourses$.add(v)
      $CurrentSemesterLoad \leftarrow CurrentSemesterLoad + v.credits$
    end if
  end for
  $Plan[t] \leftarrow SelectedCourses$ ▷ 3. 更新图结构 (Kahn 算法)
  for $v \in SelectedCourses$ **do**
    Q.remove(v)
    for $u \in$ neighbors(v) **do**
      $InDegree[u] \leftarrow InDegree[u] - 1$
      if $InDegree[u] == 0$ **then**
        Q.push(u)
      end if
    end for
  end for
  if Q is empty and all courses taken **then**
    break
  end if
end for
return $Plan$
function CalculatePriority($course, weights$) ▷ 量化指标计算
  $H \leftarrow$ LoadIndex($course$) ▷ 负荷指数
  $D \leftarrow$ Difficulty($course$) ▷ 难度系数
  $I \leftarrow$ InterestMatch($course$) ▷ 兴趣匹配度

现在，完成这些步骤的过程是朴素的，大家可以自己动手尝试！

https://github.com/MagicConch635/CourseArrangement

参考实现

https://github.com/MagicConch635/CourseArrangement

当然，更有效率的方式是...



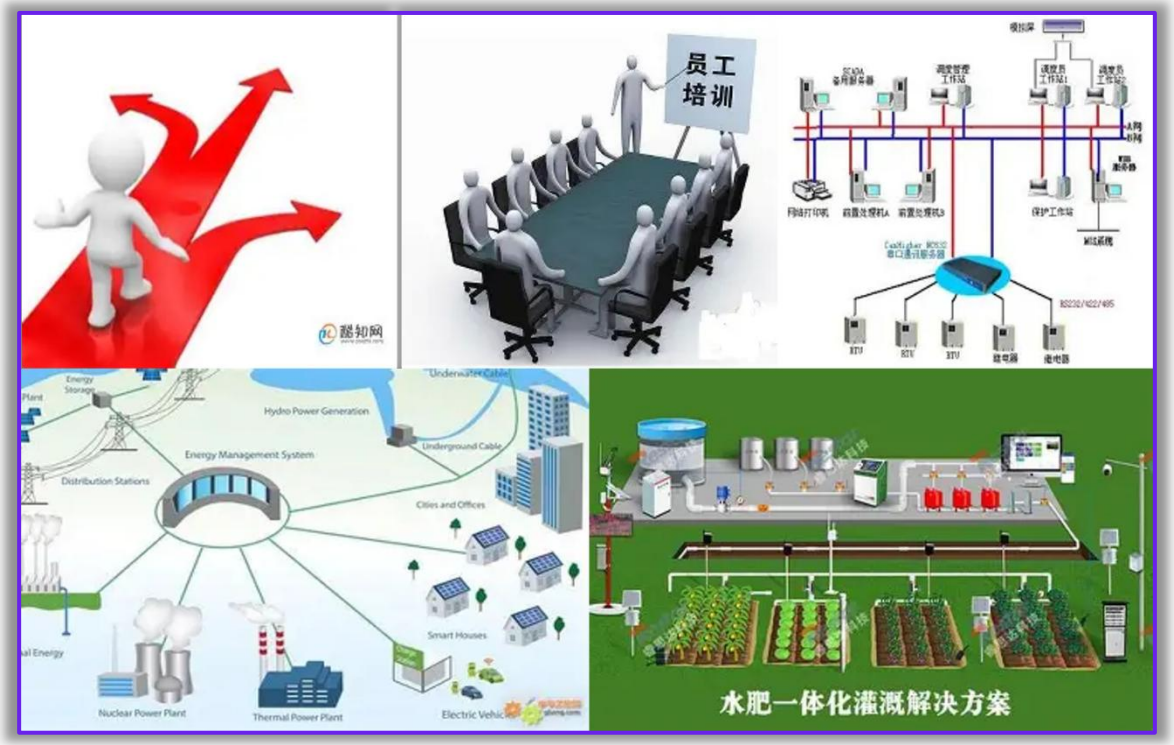
04

拓展思考

不仅仅是选课

从选课到更广泛的决策系统

- 个人技能发展规划
- 优化人力培训成本
- 智能化的电网调度
- 农业水肥管理
- 灾情物资分配
- ...



从这里，你应该带走的...

- 图结构，Kahn算法
- 如何将需求量化为可计算的指标
- 了解基本的优化方式
- 学会自己写一个排课系统！
- 以及，领悟这整个过程背后的思考方式

清华大学
Tsinghua University

感 谢 倾 听

敬请批评指正

小组成员：赵乐毅 刘宪武 卢绪涵

2025年12月30日

“运筹选律，智启未来”

小组分工

灵感提供	赵乐毅
报告初稿	赵乐毅
实现代码	刘宪武
报告终稿	卢绪涵
展示PPT	刘宪武