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 $u \in \text{neighbors}(v)$ **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
    $H \leftarrow \text{LoadIndex}(course)$ ▷ 负荷指数
    $D \leftarrow \text{Difficulty}(course)$ ▷ 难度系数
    $I \leftarrow \text{InterestMatch}(course)$ ▷ 兴趣匹配度
    return $weights.\alpha \cdot I - weights.\beta \cdot H - weights.\gamma \cdot D$
end function