本章作业（网络最优化）

- **作业 2.1:** 使用Dijkstra算法计算设备更新问题（图154）的最短路。
- **作业 2.2:** 通过构造说明最小成本流问题作为其特殊情况包含：最短路问题和最大流问题。
- **作业 2.3:** 考虑一个公司希望在员工与任务之间进行有效的分配。每个任务都需要特定的技能，并且每个员工都有一套技能。每个员工都有能处理的任务数的上限，每个任务需要一个员工去完成。
  给定数据：
  - 员工集合 $E = \{e_1, e_2, e_3\}$
  - 任务集合 $T = \{t_1, t_2, t_3, t_4\}$
  - 每个员工可以处理的任务数量为 $C = \{2, 1, 2\}$
  - 任务分配矩阵 A 定义为：

$$A = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

  其中 $A_{ij} = 1$ 表示员工 e_i 可以执行任务 t_j。

  1. 描述上述数据的流网络图。
  2. 使用Ford-Fulkerson算法或其他最大流算法，求出可以分配的最大任务数量。
  3. 指出哪个员工应该分配哪个任务以达到最大流量。
  4. 如果添加了一个新任务，它可以被 e_1 和 e_3 完成，应如何修改流网络？请指出修改后的最大流量分配。
- **Project 1.1[可选]:** 编程实现求解线性规划问题的单纯形算法，要求提交程序代码，用户指南及测试报告,测试报告需提供退化基解的相应求解结果。

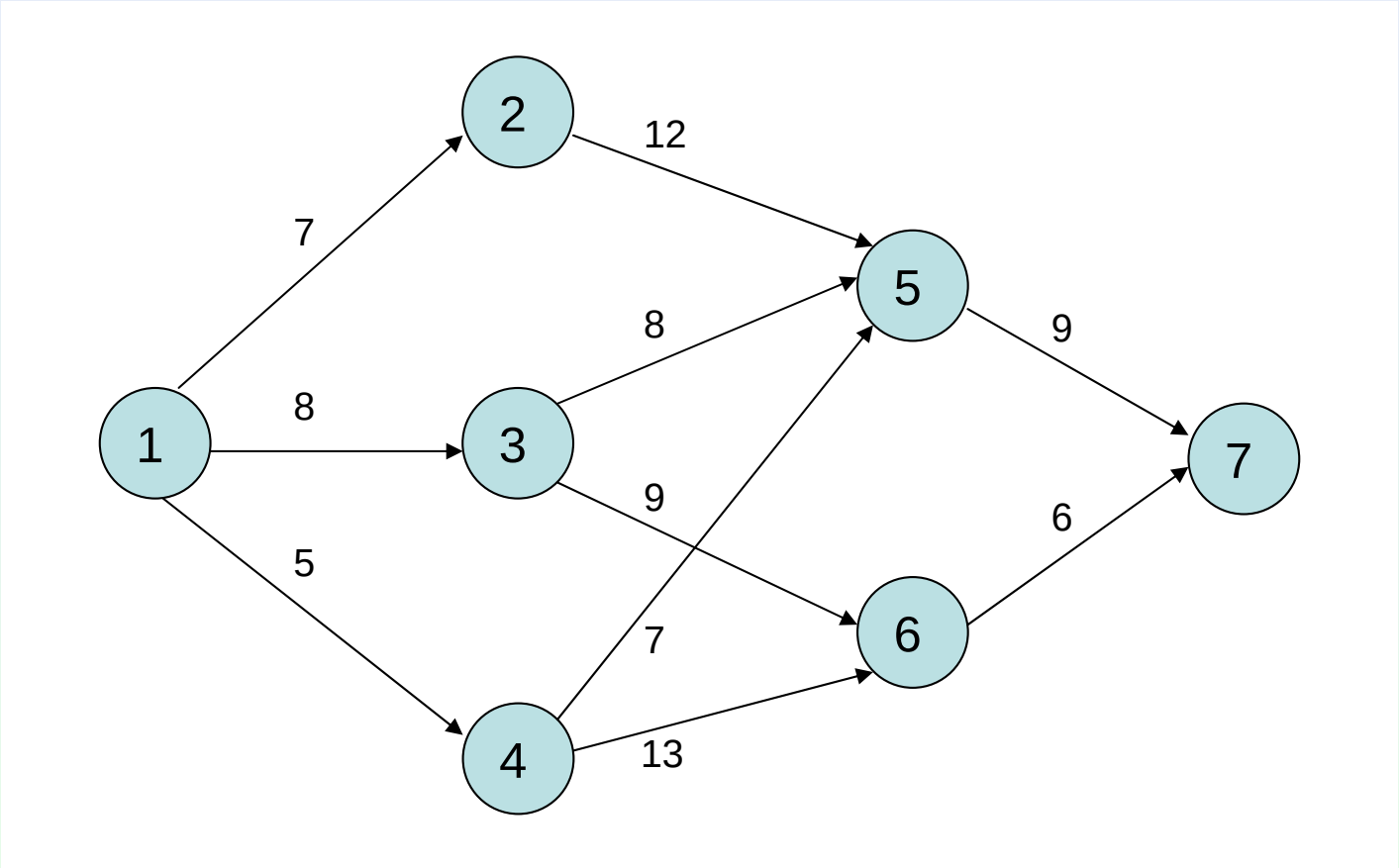
第四章 动态规划

动态规划研究的是“决策过程的最优化”，具有广泛应用背景，并已建立了严密的理论基础。

动态规划主要思想：将问题分解为子问题，并重复使用已有的结论（即写出递归式）。

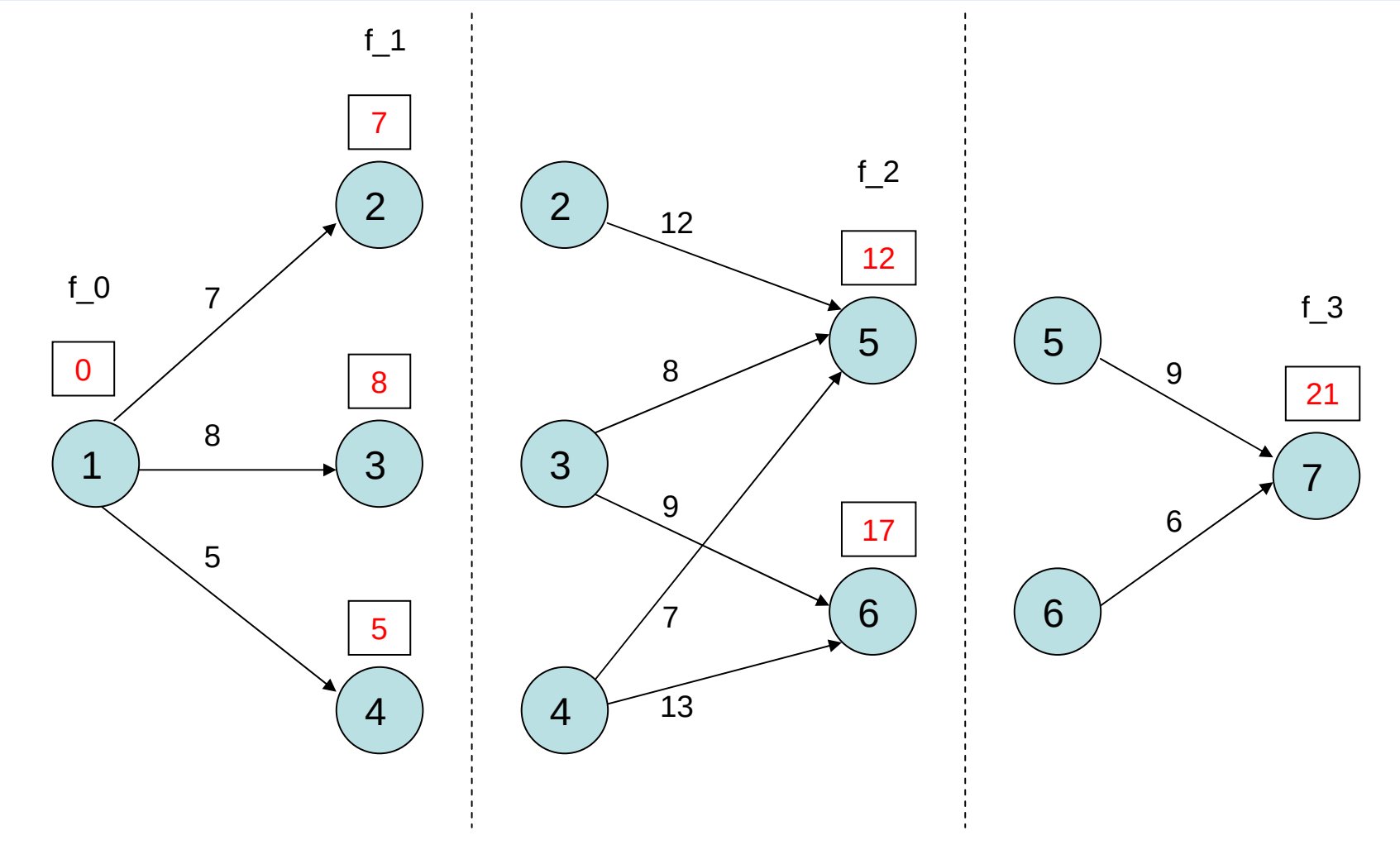
动态规划

最短路径问题的一个例子



动态规划

分阶段考虑



动态规划

递归方程(Recursive Equation)

$$\begin{cases} f_i(x_i) = \min\limits_{(x_{i-1},x_i)\in E}\{d(x_{i-1},x_i) + f_{i-1}(x_{i-1})\}, & i = 1,2,3, \\ f_0(x_0) = 0. \end{cases}$$

动态规划

动态规划的几个关键要素：

- Stages 阶段
- Alternatives 选择
- States 状态
- Recursive Equations 递归方程、状态转移方程

动态规划

背包/货物装载(Knapsack/Cargo-Loading)
01背包问题： 最基本的背包问题就是01背包问题（01 knapsack problem）：一共有n件物品，第i（i从1开始）件物品的重量为w_i，价值为r_i。在总重量不超过背包承载上限W的情况下，能够装入背包的最大价值是多少？

我们可以使用枚举法，将所有的情况列举出来，最多有2^n种情况。
也可以写成如下的整数线性规划(Integer Linear Programming)：

$$\begin{aligned} \max \quad & z = \sum_{i=1}^{n} r_i m_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i m_i \leq W \\ & m_1, \cdots, m_n \in \{0, 1\} \end{aligned}$$

完全背包问题： 完全背包问题，就是每个物品可以有无穷个。
A general (n-Items, W-LB) knapsack problem can be represented by the following Integer Linear Programming:

$$\begin{aligned} \max \quad & z = \sum_{i=1}^{n} r_i m_i \\ \text{s.t.} \quad & \sum_{i=1}^{n} w_i m_i \leq W \\ & m_1, \cdots, m_n \in \mathbb{Z}_+ \cup \{0\} \end{aligned}$$

在最差情况下，求解整数线性规划需要指数时间。

01背包问题

对于01背包问题，我们有如下动态规划解法：

- stage: 在0/1背包问题的动态规划中，阶段表示问题求解过程中的决策点或迭代。每个阶段对应于选择将要放入背包或拒绝的一项物品。通常，阶段的数量等于要考虑的物品数量。例如，如果有5个物品，动态规划过程将有5个阶段。
- state：状态表示在每个阶段定义问题所需的信息。在0/1背包问题中，每个阶段的状态通常包括以下组件：
  1. 当前考虑的物品（例如，物品i）。
  2. 在该阶段背包的剩余容量（还可以添加多少重量到背包中）。

  因此，状态可以表示为一对（i, w），其中'i'是当前物品的索引，'w'是该阶段背包的剩余容量。状态有助于跟踪子问题及其解决方案，随着阶段的进行而不断更新。
- alternative: 选择：选择指的是每个阶段可以做出的选择或决策。在0/1背包问题中，每个阶段有两种选择：
  1. 选择当前物品并将其添加到背包中，前提是其重量不超过剩余容量。
  2. 拒绝当前物品，继续下一个物品，不将其添加到背包中。

01背包问题

我们用$f(i,w)$ 表示前i个物品，放入最大承重 w的背包最大价值。我们还有$f(i,0)=0, i=0,1,...,n.$ 那么，根据上述分析，对于$n \geq i > 1, W \geq w \geq 0$,我们有如下递归方程：

$$f(i,w) = \max\{f(i-1,w), f(i-1,w-w_i)+r_i, (w \geq w_i)\}$$

其中，$f(i-1,w)$ 表示不放入物品i的价值； $f(i-1,w-w_i)+r_i$表示放入i的价值。 使用动态规划可以将复杂度降至$\mathcal{O}(nW).$

注：背包问题的判定形式（即是否能找到放入方式，使得不超过承重W的情况下达到价值V?）是NP-complete问题。对于一个数W，需要m=log W的位数来表示.因此，m才是输入规模的一部分。所以$O(n*W) = O(n2^m)$,所以是NP问题。

完全背包问题

我们仍用$f(i,w)$ 表示前i个物品，放入最大承重 w的背包最大价值。
01背包只有两种情况即取0件和取1件，而这里是取0件、1件、2件...直到超过限重（$k > w/w_i$）

1. Stage i is represented by item i, $i = 1, 2, \cdots, n$.
2. The alternatives at stage i are represented by m_i, the number of units of item i included in the knapsack. It follows that $k = 0, 1, \cdots, \lfloor \frac{w}{w_i} \rfloor$.

动态规划

我们有边界条件： $f(i,0)=0,\ i=0,1,...,n,$ 递归方程如下

$$f(i,w)=\max_{k=0,1,\cdots,\lfloor \frac{w}{w_i} \rfloor}\{r_i k+f(i-1,w-w_i k)\},\quad i=1,\cdots,n,1\leq w\leq$$

动态规划

例子： 7-ton Vessel

Item i	1	2	3
w_i	2	3	1
r_i	31	47	15

动态规划

设备更新模型(Equipment Replacement Model)

假设我们考虑的是一个跨度为n年的设备更新问题。每年初我们需要决定是否保留当前设备再使用一年或者更换一个新的设备。令$r(t)$, $c(t)$, $s(t)$分别表示一台 t-年龄设备的年营业收入，年运营成本及其残余价值。另外，在规划期内的任何一年购置一台新设备的成本是I.

动态规划

设备更新模型：设某公司现有一台3年龄的设备，需制定一个未来4年($n = 5$)的设备更新最优策略。该公司还规定6年龄的设备必须得以更换。一台新设备的成本是\$100,000. 下表给出的是设备更新问题的相关数据，其中t是机器年龄， $r(t), c(t), s(t)$ 分别表示t年龄机器的年营业收入，年运营成本及残余价值。

Table: 设备更新问题数据表

t	$r(t)$	$c(t)$	$s(t)$
0	20,000	200	–
1	19,000	600	80,000
2	18,500	1,200	60,000
3	17,200	1,500	50,000
4	15,500	1,700	30,000
5	14,000	1,800	10,000
6	12,200	2,200	5,000

动态规划

设备更新模型

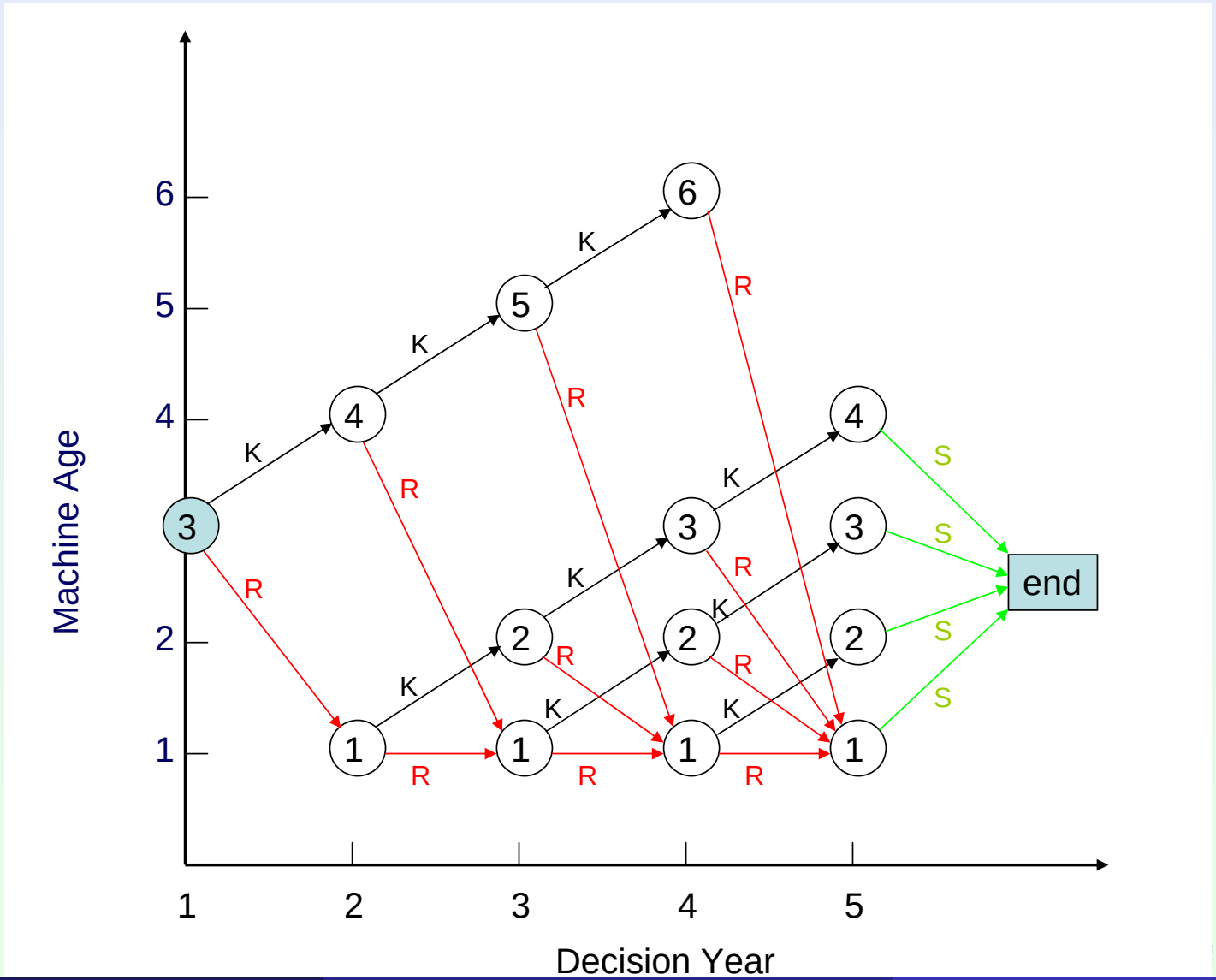
- Stage i is represented by year i where $i = 1, \cdots, n$.
- The alternatives at stage i are either *keeping* or *replacing* the machine at the start of year i.
- The state at stage i is the age of the machine at the start of year i.
- Recursive equation:

$$f_i(t) = \max \begin{cases} r(t) - c(t) + f_{i+1}(t+1), t \leq 6 & \text{if Keeping} \\ r(0) - c(0) + s(t) - I + f_{i+1}(1), & \text{if Replacing} \end{cases}$$
$$f_{n+1}(t) = s(t)$$

where $f_i(t)$ is defined as the maximum net income for years $i, i+1, \cdots, n$ by given a t-year-old machine at the start of year i.

动态规划

设备更新模型



本章作业（动态规划）

- **Exercise 3.1:** 设现有一台2年龄的设备，另规定5年龄的设备必须更换。在规划期购置新设备的成本分别是

$$(p_1, p_2, p_3, p_4, p_5) = (100, 105, 110, 115, 120).$$

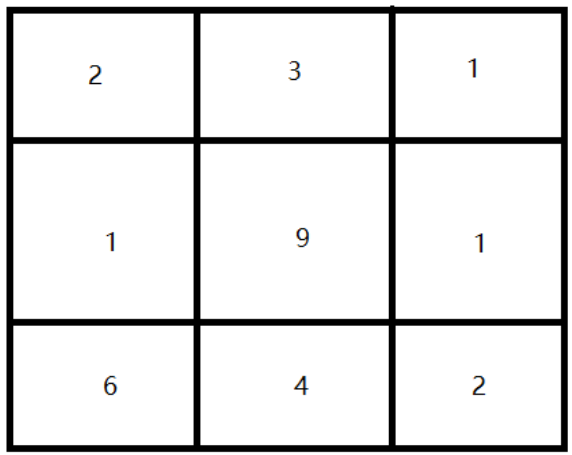
试构建如下设备更新的动态规划模型并求其最优更新策略。

Table: 五年期设备更新

设备年龄t	残余价值v_t	运行费用c_t
0	-	30
1	50	40
2	25	50
3	10	75
4	5	90
5	2	-

本章作业（动态规划）

- **Exercise 3.2:**
  问题1：有一个移动机器人，可以在二维矩阵平面中移动，其移动的起点位于左上角，每次移动只能向右或者向下，其移动的终点是右下角，在这个矩阵形的平面中每个位置都有一个数值，代表机器人在经过这个区域时需要付出的代价。我们的目标就是，在这个平面中，找一条代价最小的路径，并且给出最小代价路径。
  问题2：若机器人从左上角出发，起点坐标设为（0，0），需要达到右下角，坐标设为（M，N）。请问有多少种不同的路径（无需考虑每格代价）？



本章作业（动态规划）

- **Project 3.3[可选]:** 编程实现最短路径算法，要求提交程序代码，用户指南及测试报告,测试报告需报告算法的求解时间随着问题规模变化情况。
  第二次作业包括第二章3道习题和本章2道习题，提交时间为10月22日24点之前。共有3个project题目，任选2个完成，学期结束前2周提交。