**西安电子科技大学**

**算法设计与分析 课程实验报告**

计算机科学与技术 学院 1903052 班

**实验名称**

姓名 赵书晨 学号 19030500122

实验日期 2022 年 4 月 6 日

### 一、诚信声明

本次实验为本人独立完成，编程语言使用Javascript。

### 二、实验内容

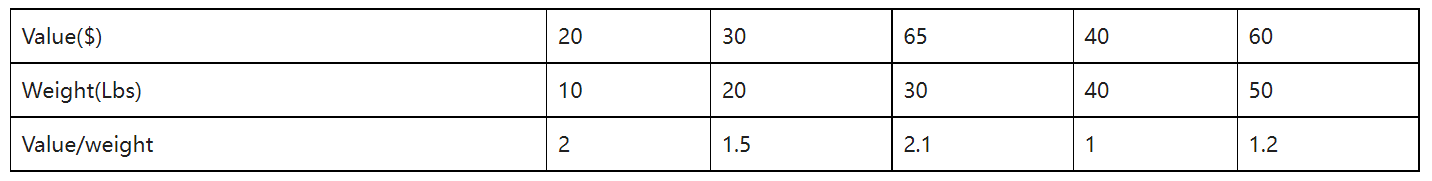
##### （一）背包问题

可分背包采用贪心策略，优先放入性价比最高的。

0-1背包采用动态规划，状态转移方程为：

dp[i][v] = Math.max(dp[i-1][v], dp[i-1][v-w[i]]+value[i])

同时可以采用滚动数组的方法，进一步优化空间复杂度。



可分背包结果：163

0-1背包结果：155

##### （二）调度算法

依旧是一个贪心问题，如果要平均完成时间最短，则采用短作业优先调度算法；

又因为是非抢占式调度，所以先按作业到达时间排序，如果同时到达则优先处理需要时间短的。

let jobs = [ [15, 0], [8, 3], [3, 50], [10, 3], [3, 1] ]; //need time, arrive time

调度顺序：[ 15, 0 ], [ 3, 1 ], [ 8, 3 ], [ 10, 3 ], [ 3, 50 ]

平均完成时间：53/5 = 10.6

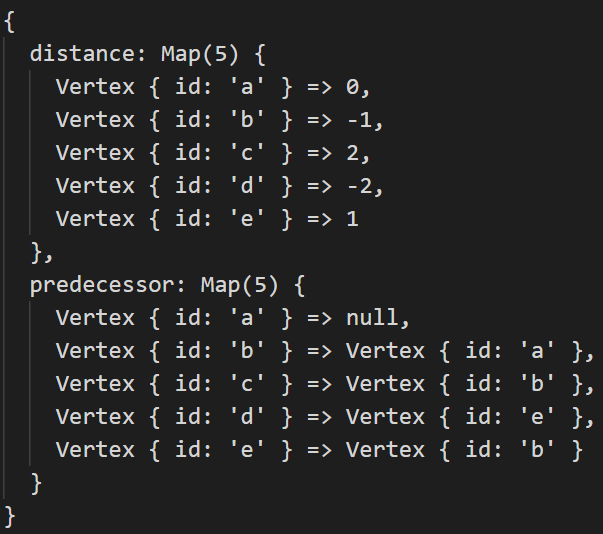
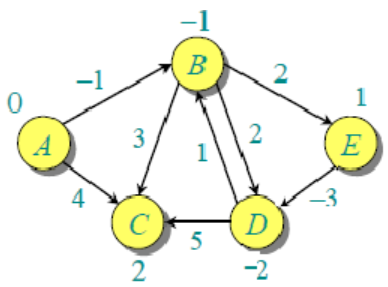
##### （三）单源最短路径

因为给出的邻接矩阵存在负权重的边，所以使用Bellman-Ford算法。

其中关键的松弛函数：判断是否经过某个顶点/某条边，可以缩短起点到终点的路径权值。

对边集合E中任意边，以w(u,v)表示顶点u出发到顶点v的边的权值，以d[v]表示当前从起点s到顶点v的路径权值。

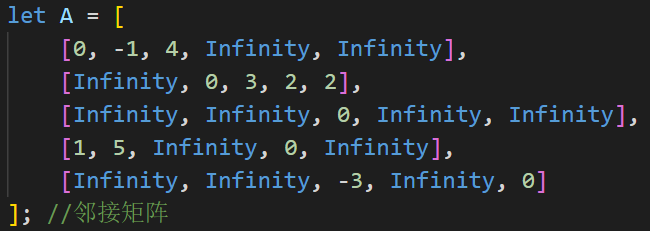
若存在边w(u,v)，使得：d[v] > d[u] + w(u,v)，则更新d[v]值：d[v] =d[u] + w(u,v)。



##### （四）全部最短路径

使用Floyd算法。

对于每个顶点v，和任一顶点对(i,j)，i != j，v != i，v != j，若A[i][j] > A[i][v] + A[v][j]，则将A[i][j] 更新为A[i][v] + A[v][j]的值，并将Path[i][j]改为v。



V0-V1 weight: -1 Path: 0 -> 1

V0-V2 weight: -2 Path: 0 -> 1 -> 4 -> 2

V0-V3 weight: 1 Path: 0 -> 1 -> 3

V0-V4 weight: 1 Path: 0 -> 1 -> 4

V1-V2 weight: -1 Path: 1 -> 4 -> 2

V1-V3 weight: 2 Path: 1 -> 3

V1-V4 weight: 2 Path: 1 -> 4

V2-V3 weight: Infinity Path: 2 -> 3

V2-V4 weight: Infinity Path: 2 -> 4

V3-V4 weight: 2 Path: 3 -> 0 -> 1 -> 4

### 三、实验中遇到的问题

1. 状态转移方程

2. 滚动数组

3. 图的数据结构存储

4. 松弛函数

### 四、实验小结

通过本次实验，了解并掌握了动态规划及贪心策略，对图中的最短路径问题有了进一步的认识。