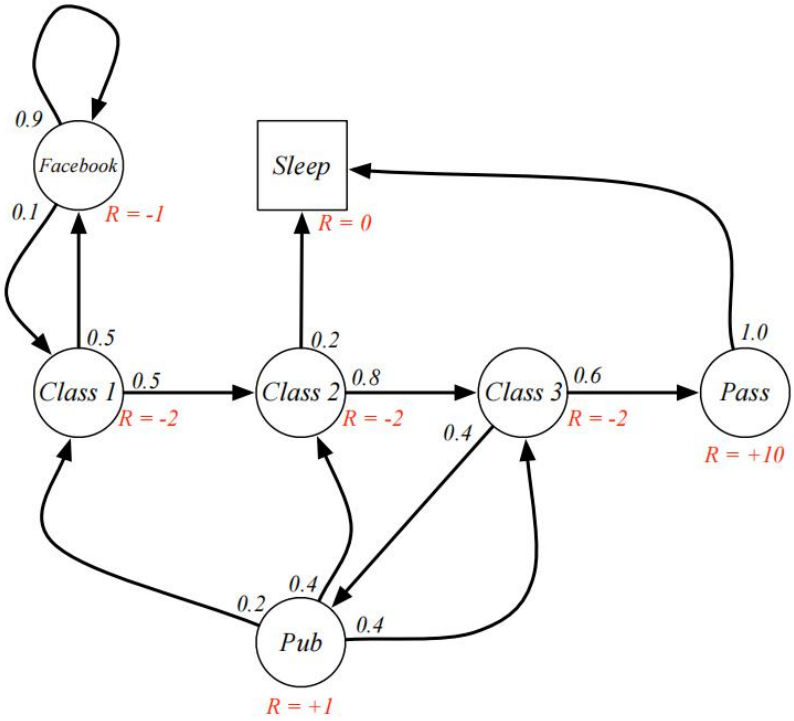
问题：用数值方法解线性方程组解决以下学生马尔可夫收益过程



代码：

```
# 迭代
def next_v(_lambda, r, v0, P):
    v = []
    for j in range(len(v0)):
        if j != len(v0) - 1:
            exp = .0
            # 与当前状态相接的后续状态的值函数相应的状态转移概率的求和的折扣
            for k in range(len(P[j])):
                exp += P[j][k][0] * v0[P[j][k][1]]
            # 当前状态在下一阶段的报酬
            v.append(r[j] + _lambda * exp)
    # Sleep状态无后续状态，故直接赋值0
    v.append(0.0)
    return v

if __name__ == '__main__':
    # γ
    _lambda = 1
    # 报酬顺序：Class 1、Class 2、Class 3、Pass、Pub、Facebook、Sleep，分别对应0, 1, 2, 3, 4, 5, 6
```

```
    R = [-2., -2., -2., 10., 1., -1., 0.]
    # 初始化值函数
    v0 = [0, 0, 0, 0, 0, 0, 0]
    # 状态转移概率
    P = [[[0.5, 1], [0.5, 5]],
                      [[0.8, 2], [0.2, 6]],
                      [[0.6, 3], [0.4, 4]],
                      [[1, 6]],
                      [[0.2, 0], [0.4, 1], [0.4, 2]],
                      [[0.1, 0], [0.9, 5]],
                      [[0, 0]]]
    v = []
    # 指定迭代次数
    for i in range(1000):
        v = next_v(_lambda, R, v0, P)
        # 用新生成的值函数列表替换旧的值函数列表
        v0 = v
    print(v)
```

在构造转移矩阵的时候，可以直接构造矩阵，此处只是为了方便

结果：

```
[-12.543209876543184, 1.456790123456793, 4.320987654320991, 10.0, 0.8024691358024769, -22.543209876543163, 0.0]
```