

例 1.18 Python 程序代码

$V_e(v)$:最早发生时间: 是指从始点开始到顶点 V_k 的最大路径长度

$V_l(v)$:最迟发生时间:在不推迟整个工期的前提下, 事件 v_k 允许的最晚发生时间

$e(v)$:最早发生时间:某工序最早开始时间

$l(v)$:最迟发生时间: 某工序最晚开始时间

$d(v)$:活动时间余量: $d(v) = l(v) - e(v)$

"""

```
def find_key_path(node_list, edge_list):
```

```
    #顶点列表
```

```
    ''' 计算各个顶点的  $V_e(v)$  最早发生时间'''
```

```
    #找出图的起点
```

```
    temp_start_list = []
```

```
    for edge in edge_list:
```

```
        temp_start_list.append(edge[1])
```

```
    start_node = [x for x in node_list if x not in temp_start_list]
```

```
    # print(start_node)
```

```
    Ve_node_dict = {}
```

```
    Ve_node_dict[start_node[0]] = 0
```

```
    for node in node_list:
```

```
        Ve_templine_list = []
```

```
        for edge in edge_list:
```

```
            if node == edge[1]:
```

```
                temp_Ve_node_value = Ve_node_dict[edge[0]] + edge[2]
```

```
                Ve_templine_list.append(temp_Ve_node_value)
```

```
        if len(Ve_templine_list) == 0:
```

```
            Ve_node_dict[node] = 0
```

```
        if len(Ve_templine_list) == 1:
```

```
            Ve_node_value = Ve_templine_list[0]
```

```
            Ve_node_dict[node] = Ve_node_value
```

```
        if len(Ve_templine_list) > 1:
```

```
            Ve_node_value = max(Ve_templine_list)
```

```
            Ve_node_dict[node] = Ve_node_value
```

```
    print('  $V_e(v)$  最早发生时间:\n', Ve_node_dict, '\n')
```

```
    ''' 计算各个顶点的  $V_l(v)$  最迟发生时间'''
```

```
    #找出图的终点
```

```
    temp_end_list = []
```

```
    for edge in edge_list:
```

```
        temp_end_list.append(edge[0])
```

```

end_node = [x for x in node_list if x not in temp_end_list]
# print(end_node)

```

```

Vl_node_dict = {}
Vl_node_dict[end_node[0]] = Ve_node_dict[end_node[0]]

```

```

reverse_edge_list = []
for i in range(len(edge_list)-1,-1,-1):
    reverse_edge_list.append(edge_list[i])

```

```

for node in reversed(node_list):
    Vl_temptime_list = []
    for edge in reverse_edge_list:
        if node == edge[0]:
            temp_Vl_node_value = Vl_node_dict[edge[1]] - edge[2]
            Vl_temptime_list.append(temp_Vl_node_value)
    if len(Vl_temptime_list) == 0:
        Vl_node_dict[node] = Ve_node_dict[end_node[0]]
    if len(Vl_temptime_list) == 1:
        Vl_node_value = Vl_temptime_list[0]
        Vl_node_dict[node] = Vl_node_value
    if len(Vl_temptime_list) > 1:
        Vl_node_value = min(Vl_temptime_list)
        Vl_node_dict[node] = Vl_node_value
print('Vl(v) 最迟发生时间:\n', Vl_node_dict, '\n')

```

''' 计算各个边的 e(a) 最早发生时间'''

```

e_bian_dict = {}
for edge in edge_list:
    e_bian_dict[' {}-{}'.format(edge[0], edge[1])] = Ve_node_dict[edge[0]]
print('e(a) 最早发生时间:\n', e_bian_dict, '\n')

```

''' 计算各个边的 l(a) 最迟发生时间'''

```

l_bian_dict = {}
for edge in edge_list:
    l_bian_dict[' {}-{}'.format(edge[0], edge[1])] = Vl_node_dict[edge[1]] -
edge[2]
print('l(a) 最迟发生时间:\n', l_bian_dict, '\n')

```

''' 计算时间余量 d(a)'''

```

d_bian_dict = {}
for bian in e_bian_dict.keys():
    d_bian_dict[bian] = l_bian_dict[bian] - e_bian_dict[bian]
print("d(a)时间余量:\n",d_bian_dict,'\n')

print("关键路径为: ",[x for x in d_bian_dict if d_bian_dict[x] == 0])
print(f"关键路径长度为: {Vl_node_dict[end_node[0]]}")

if __name__ == "__main__":
    node_list = ['1','2','3','4','5','6','7','8','9','10']

    edge_list = [['1', '2', 3],
                  ['2', '3', 0],
                  ['2', '5', 3],
                  ['3', '4', 2],
                  ['4', '5', 0],
                  ['4', '7', 3],
                  ['5', '6', 4],
                  ['6', '7', 2],
                  ['6', '8', 0],
                  ['7', '9', 2],
                  ['8', '9', 1],
                  ['9', '10', 6],
                  ]

    find_key_path(node_list,edge_list)

```