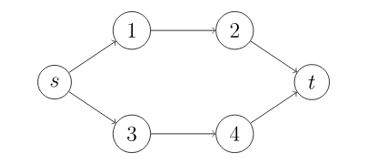
**二、问题二的模型建立与求解**

**2.1图论模型建立**

**2.1.1拆点**

问题二与问题一不同的点在于，问题一的权值是在边上的，而问题二的权值是在节点上的，即使用一次黄色石头减1分，使用一次蓝色石头减2分等，利用点权建图的难处在于，传统算法如Dijkstra并不适用于点权图，并且用多种数据结构维护同一个点集合，会导致代码过于臃肿，导致算法效率低。

对此，可将单独一个节点拆成两个节点，即原节点的出点和入点，那么原节点的出点和入点的连边即为节点的权值，至此，即可将点权转化为边权，示意图如下：



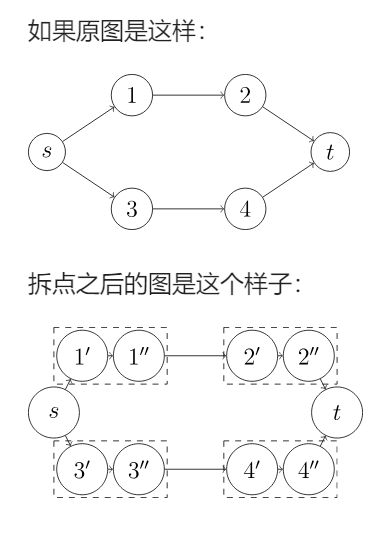


图2.1 拆点示意图

**2.1.2题意转化**

对于该节点与其他节点的连边，仍然采用图1.4中的的连线图，以保证图的大部分是连通的，并将该节点的出点与其他节点的入点相连，并设置该边的权值为0，以避免距离权值会影响到问题二中的颜色权值。

由于题目规定是采取扣分制度，那么可将问题转化为，使用一次黄色石头加1分，使用一次蓝色石头加2分，使用一次红色石头加3分，使用一次绿色石头加4分，而后再通过Dijkstra算法求取从起点到终点的最短路径，即最少得分路径，而后再通过满分1000，减去对应的最少得分，则可得到最终的最高得分。

**2.2图论模型求解**

综上分析，首先将图中所有节点均拆为2个节点即入点和出点，那么图中节点的数量就要增大2倍，而后再从入点向出点连接一条权值为原节点颜色权值的边，为了保证图的连通性，将在的图上进行，并把原距离权值赋为0，这里仍然选择编号1作为起点，而编号为111，112，113，114则作为终点，最后通过题意转化，使得只需再使用Dijkstra算法求取最短路，则可得到最高分路径。

表2.1 问题二在上的最高得分

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 每次攀爬最大高度/编号 | 111 | 112 | 113 | 114 |
| 100 | 969 | 967 | 967 | 969 |

在Dijkstra算法寻找最短路的过程中，只需要用一个数组prev[N]记录每个节点在最短路径中的前驱节点，而后再通过从终点向起点回溯，将回溯得到的路径依次添加到一个路径数组中，然后反向输出这个路径数组，就能够得到从起点到终点的具体路径，故可得到下表即为从起点编号1的节点向，编号为111，112，113，114的最高得分路径。

表2.2 问题二从起点向终点的最高得分路径

|  |  |
| --- | --- |
| 终点编号 | 路径 |
|  |  |
|  |  |
|  |  |
|  |  |

在图上可直观由下图表示：

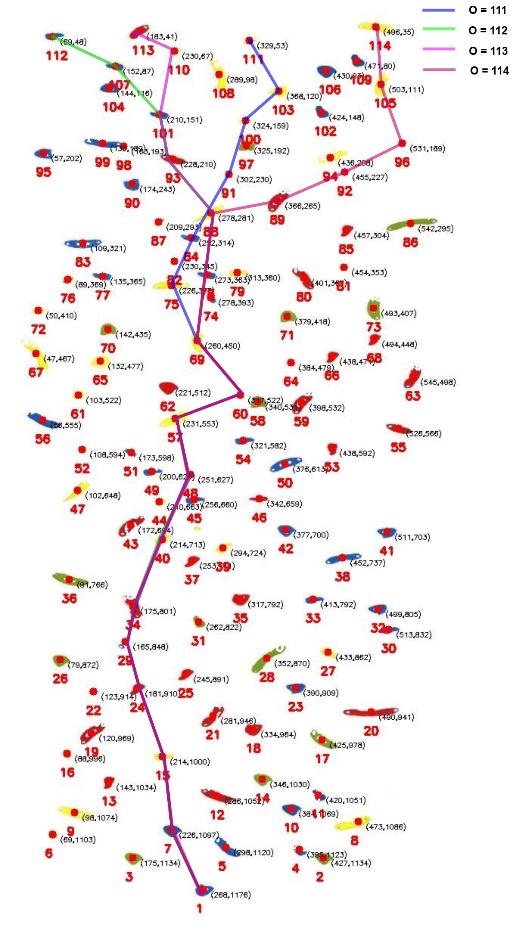


图2.2 起点向终点最高得分路径图