

图1：A*寻路

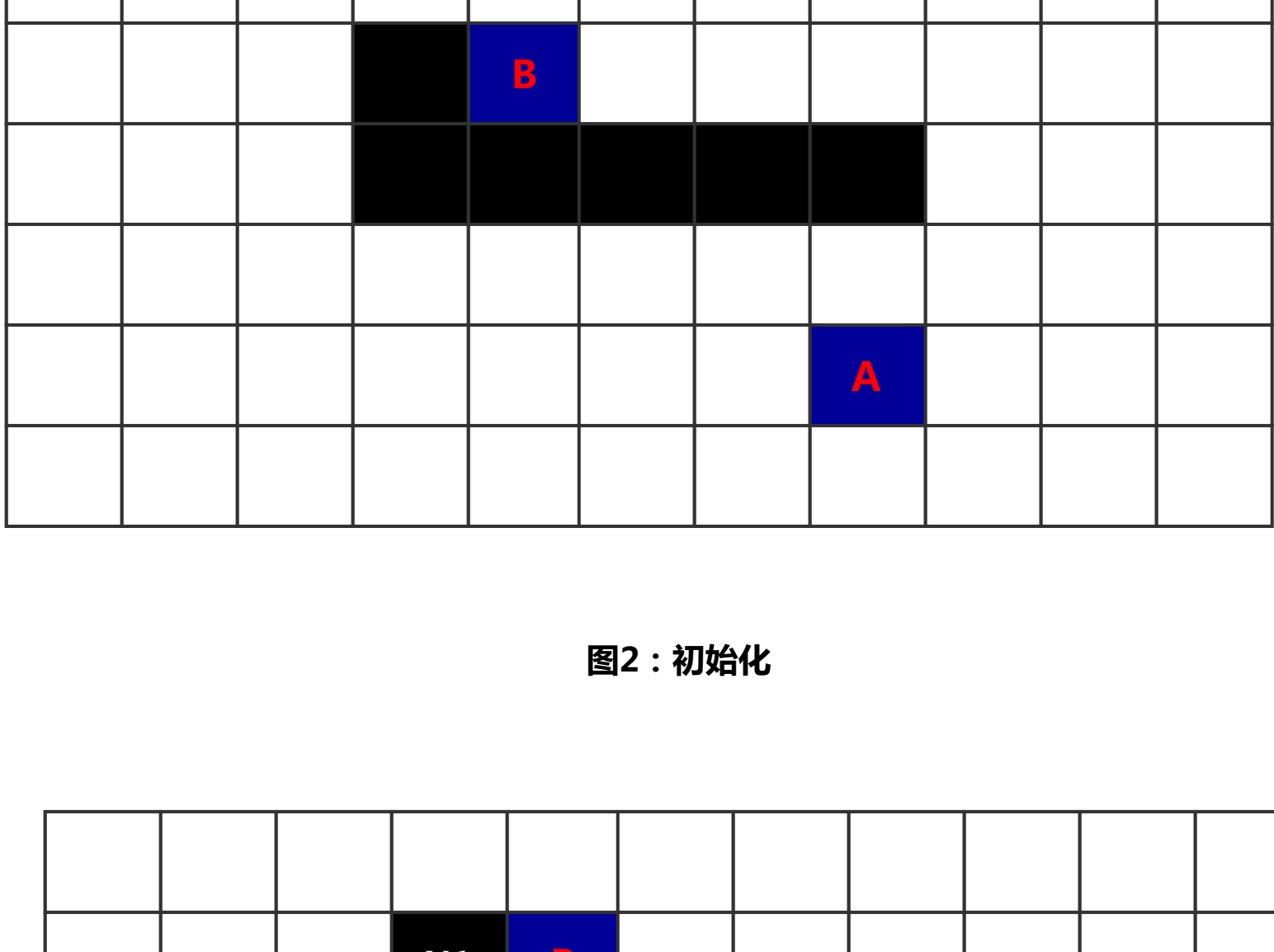
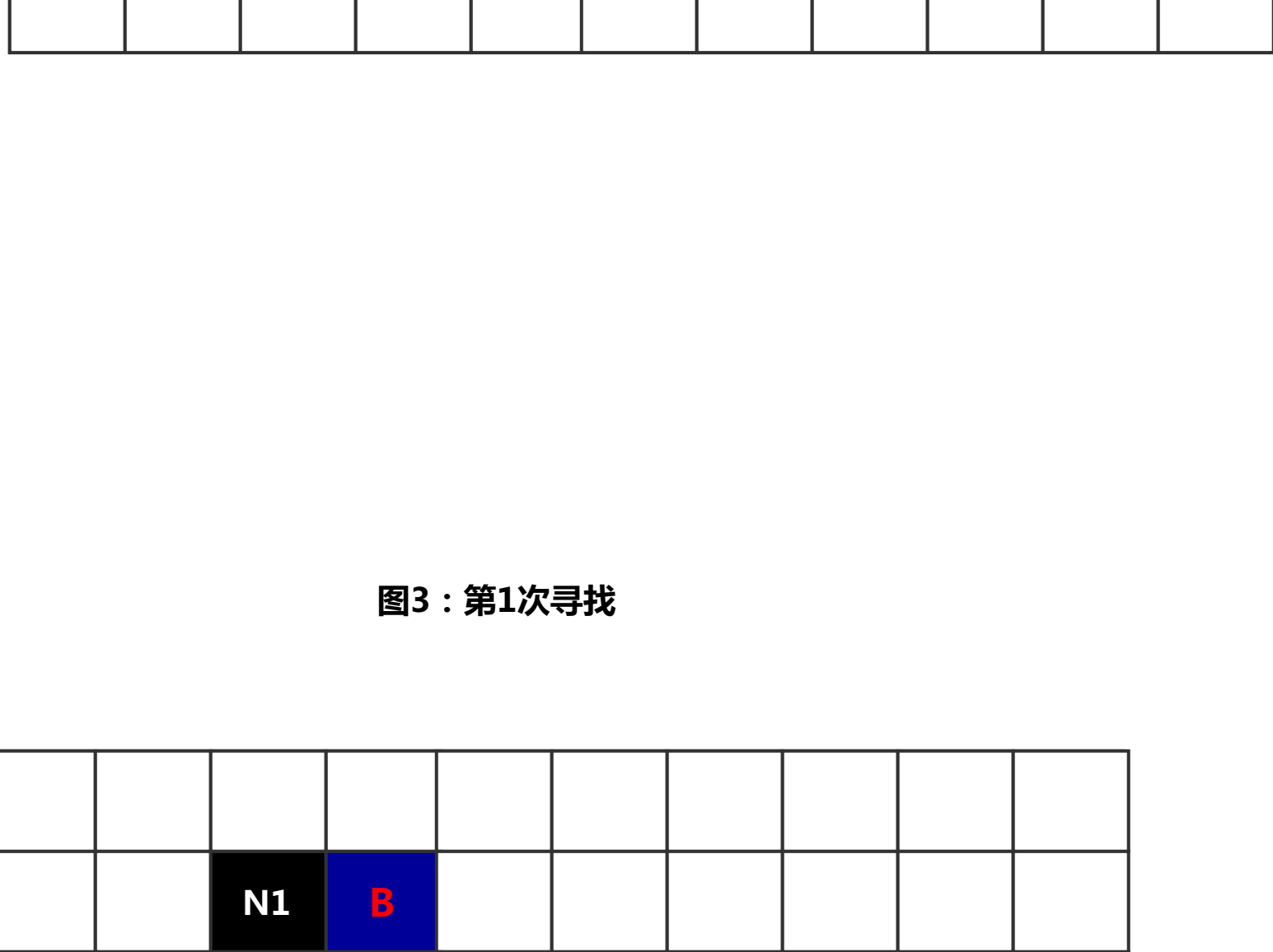


图2：初始化



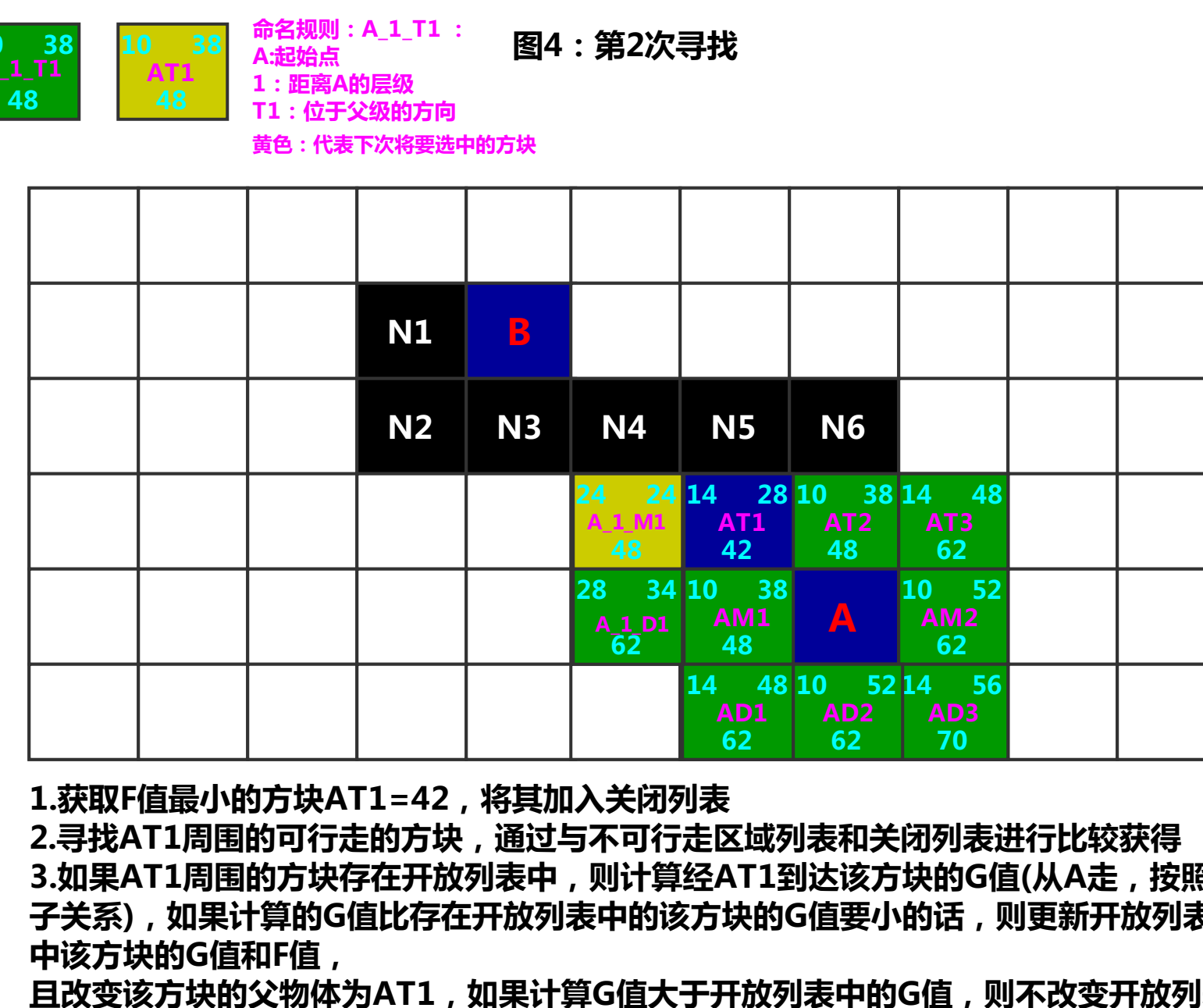
开放列表

关闭列表

不可行走区域

| 开放列表 | 关闭列表 | 不可行走区域 |
|------|------|--------|
| A | | N1 |
| | | N2 |
| | | N3 |
| | | N4 |
| | | N5 |
| | | N6 |

图3：第1次寻找



- 1.将A加入关闭列表中
- 2.寻找A周围的可行走的方块，通过与不可行走区域列表和关闭列表进行比较获得
- 3.将可行走的方块的父物体全部设为A
- 4.将可行走的方块加入开放列表，F值顺序排列

开放列表

关闭列表

不可行走区域

| 开放列表 | 关闭列表 | 不可行走区域 |
|--|------|--------|
| AT1:42 AT2:48 AM1:48 AT3:62 AD1:62 AM2:62 AD2:62 AD3:70 | A:0 | N1 |
| | | N2 |
| | | N3 |
| | | N4 |
| | | N5 |
| | | N6 |

图4：第2次寻找



- 1.获取F值最小的方块AT1=42，将其加入关闭列表
- 2.寻找AT1周围的可行走的方块，通过与不可行走区域列表和关闭列表进行比较获得
- 3.如果AT1周围的方块存在开放列表中，则计算经AT1到达该方块的G值(从A走，按照父子关系)，如果计算的G值比存在开放列表中的该方块的G值要小的话，则更新开放列表中该方块的G值和F值，且改变该方块的父物体为A
- 4.将可行走的方块加入开放列表，F值顺序排列
- 5.将可行走的方块的父物体全部设为A
- 6.选取F值最小的，如果F最小，则比价H值，选取H值最小的

此操作更新的开放的列表中的方块有：

无

开放列表

关闭列表

不可行走区域

| 开放列表 | 关闭列表 | 不可行走区域 |
|--|---------------|----------------------------------|
| A_1_M1:48 AT2:48 AM1:48 A_1_D1:62 AT3:62 AD1:62 AM2:62 AD2:62 AD3:70 | A:0 AT1:42 | N1 N2 N3 N4 N5 N6 |

图5：第3次寻找



- 1.获取F值最小的方块A_1_M1=48，将其加入关闭列表
- 2.寻找A_1_M1周围的可行走的方块，通过与不可行走区域列表和关闭列表进行比较获得
- 3.如果A_1_M1周围的方块存在开放列表中，则计算经A_1_M1到达该方块的G值(从A走，按照父子关系)，如果计算的G值比存在开放列表中的该方块的G值要小的话，则更新开放列表中该方块的G值和F值，且改变该方块的父物体为A_1_M1，如果计算G值大于开放列表中的G值，则不改变开放列表中该方块的值
- 4.将可行走的方块加入开放列表，F值顺序排列
- 5.将可行走的方块的父物体全部设为A_1_M1
- 6.选取F值最小的，如果F最小，则比价H值，选取H值最小的

此操作更新的开放的列表中的方块有：

A_2_D1:68

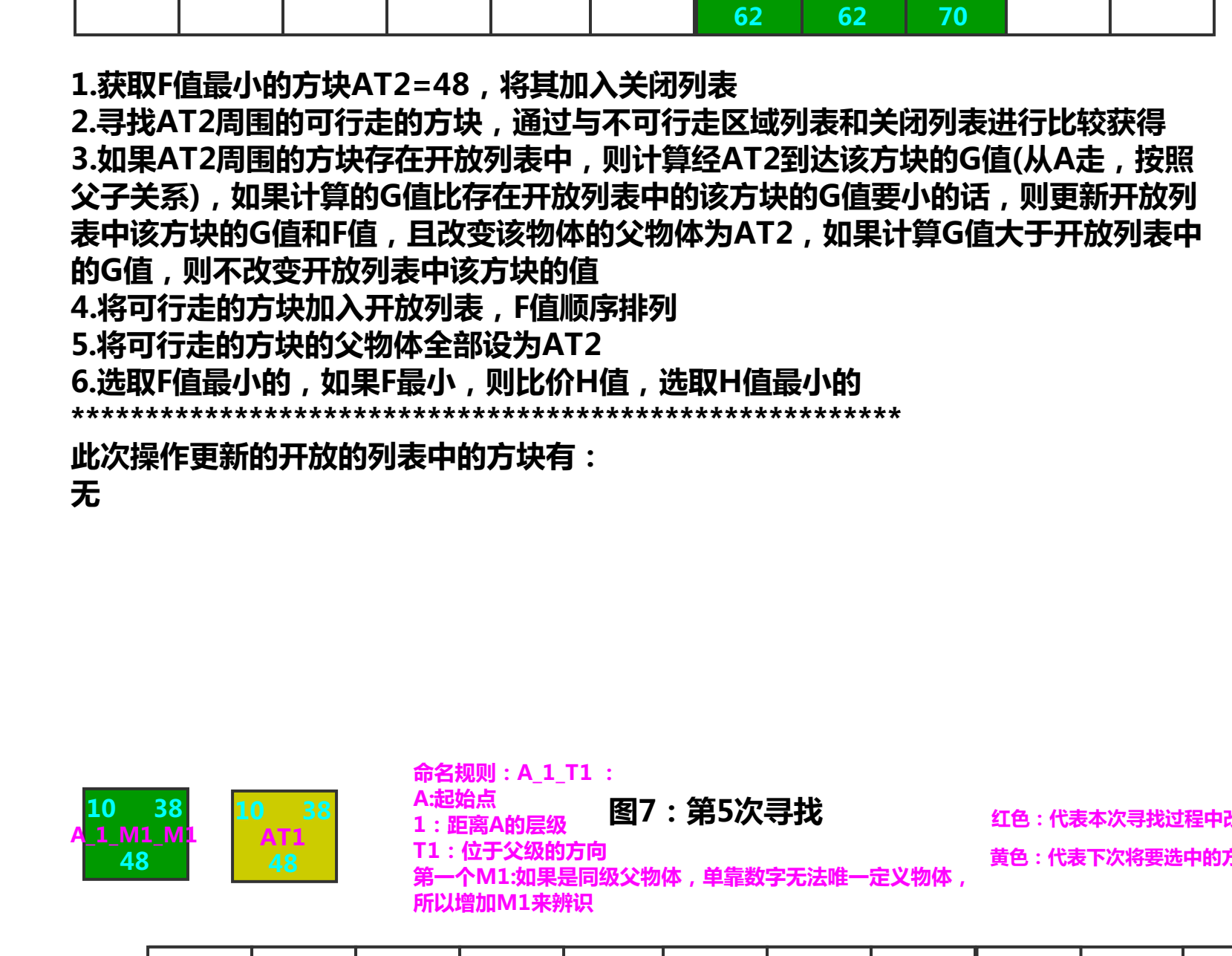
开放列表

关闭列表

不可行走区域

| 开放列表 | 关闭列表 | 不可行走区域 |
|---|----------------------------|----------------------------------|
| AT2:48 AM1:48 A_2_M1:54 A_1_D1:62 AT3:62 AD1:62 AM2:62 AD2:62 A_2_D1:68 AD3:70 | A:0 AT1:42 A_1_M1:48 | N1 N2 N3 N4 N5 N6 |

图6：第4次寻找



- 1.获取F值最小的方块AT2=48，将其加入关闭列表
- 2.寻找AT2周围的可行走的方块，通过与不可行走区域列表和关闭列表进行比较获得
- 3.如果AT2周围的方块存在开放列表中，则计算经AT2到达该方块的G值(从A走，按照父子关系)，如果计算的G值比存在开放列表中的该方块的G值要小的话，则更新开放列表中该方块的G值和F值，且改变该方块的父物体为AT2，如果计算G值大于开放列表中的G值，则不改变开放列表中该方块的值
- 4.将可行走的方块加入开放列表，F值顺序排列
- 5.将可行走的方块的父物体全部设为AT2
- 6.选取F值最小的，如果F最小，则比价H值，选取H值最小的

此操作更新的开放的列表中的方块有：

无

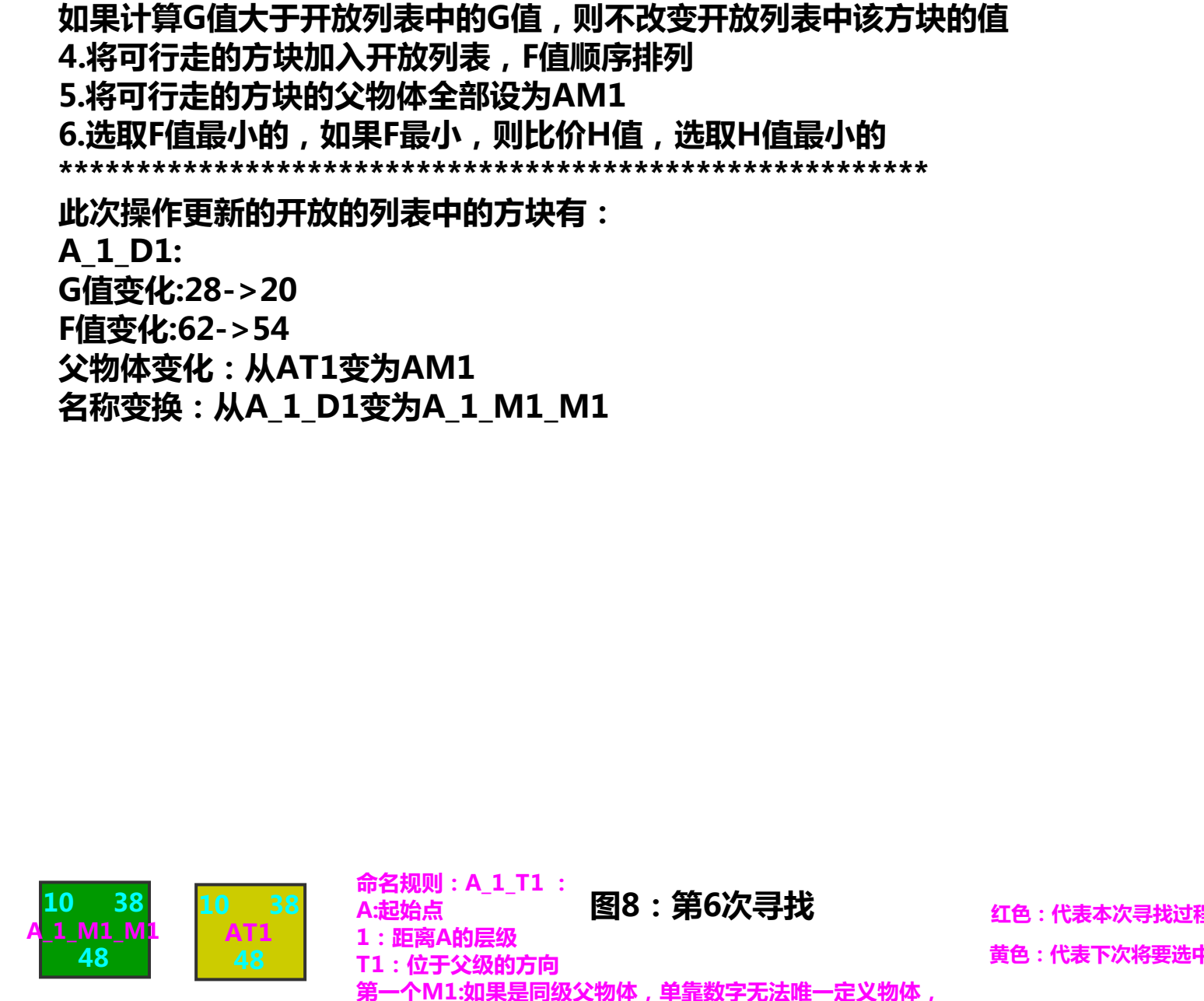
开放列表

关闭列表

不可行走区域

| 开放列表 | 关闭列表 | 不可行走区域 |
|---|--|----------------------------------|
| AM1:48 A_2_M1:54 A_1_D1:62 AT3:62 AD1:62 AM2:62 AD2:62 A_1_T2_T3:68 A_2_D1:68 AD3:70 | A:0 AT1:42 A_1_M1:48 AT2:48 AM1:48 | N1 N2 N3 N4 N5 N6 |

图7：第5次寻找



- 1.获取F值最小的方块AM1=48，将其加入关闭列表
- 2.寻找AM1周围的可行走的方块，通过与不可行走区域列表和关闭列表进行比较获得
- 3.如果AM1周围的方块存在开放列表中，则计算经AM1到达该方块的G值(从A走，按照父子关系)，如果计算的G值比存在开放列表中的该方块的G值要小的话，则更新开放列表中该方块的G值和F值，且改变该方块的父物体为AM1
- 4.将可行走的方块加入开放列表，F值顺序排列
- 5.将可行走的方块的父物体全部设为AM1
- 6.选取F值最小的，如果F最小，则比价H值，选取H值最小的

此操作更新的开放的列表中的方块有：

A_1_D1:
G值变化:28->20
F值变化:62->54
父物体变化:从AT1变为AM1
名称变换:从A_1_D1变为A_1_M1_M1

开放列表

关闭列表

不可行走区域

| 开放列表 | 关闭列表 | 不可行走区域 |
|--|--|----------------------------------|
| A_1_M1_M1:54 A_2_M1:54 A_1_M1_T1:62 AT3:62 AD1:62 AM2:62 AD2:62 A_1_T2_T3:68 A_2_D1:68 AD3:70 | A:0 AT1:42 A_1_M1:48 AT2:48 AM1:48 | N1 N2 N3 N4 N5 N6 |

图8：第6次寻找



- 1.获取F值最小的方块A_1_M1_M1=54，将其加入关闭列表
- 2.寻找A_1_M1_M1周围的可行走的方块，通过与不可行走区域列表和关闭列表进行比较获得
- 3.如果A_1_M1_M1周围的方块存在开放列表中，则计算经A_1_M1_M1到达该方块的G值(从A走，按照父子关系)，如果计算的G值比存在开放列表中的该方块的G值要小的话，则更新开放列表中该方块的G值和F值，且改变该方块的父物体为A_1_M1_M1
- 4.将可行走的方块加入开放列表，F值顺序排列
- 5.将可行走的方块的父物体全部设为A_1_M1_M1
- 6.选取F值最小的，如果F最小，则比价H值，选取H值最小的

此操作更新的开放的列表中的方块有：

A_2_D1:
G值变化:38->30
F值变化:68->60
父物体变化:从A_1_M1变为A_2_M1_M1
名称变换:从A_2_D1变为A_2_M1_M1

开放列表

关闭列表

不可行走区域

| 开放列表 | 关闭列表 | 不可行走区域 |
|--|---|----------------------------------|
| A_2_M1_M1:60 A_1_M1_T1:62 AT3:62 AD1:62 AM2:62 AD2:62 A_1_T2_T3:68 A_2_M1_D1:74 A_2_M1_D1:74 | A:0 AT1:42 A_1_M1:48 AT2:48 AM1:48 A_1_M1_M1:54 A_2_M1:54 | N1 N2 N3 N4 N5 N6 |

图9：第7次寻找



- 1.获取F值最小的方块A_2_M1_M1=60，将其加入关闭列表
- 2.寻找A_2_M1_M1周围的可行走的方块，通过与不可行走区域列表和关闭列表进行比较获得
- 3.如果A_2_M1_M1周围的方块存在开放列表中，则计算经A_2_M1_M1到达该方块的G值(从A走，按照父子关系)，如果计算的G值比存在开放列表中的该方块的G值要小的话，则更新开放列表中该方块的G值和F值，且改变该方块的父物体为A_2_M1_M1
- 4.将可行走的方块加入开放列表，F值顺序排列
- 5.将可行走的方块的父物体全部设为A_2_M1_M1
- 6.选取F值最小的，如果F最小，则比价H值，选取H值最小的

此操作更新的开放的列表中的方块有：

A_3_M1_D1:
G值变化:48->40
F值变化:82->74
父物体变化:从A_2_M1变为A_3_M1_M1

开放列表

关闭列表

不可行走区域

| 开放列表 | 关闭列表 | 不可行走区域 |
|--|---|----------------------------------|
| A_1_M1_T1:62 AT3:62 AD1:62 AM2:62 AD2:62 A_1_T2_T3:68 A_2_M1_D1:74 A_2_M1_D1:74 | A:0 AT1:42 A_1_M1:48 AT2:48 AM1:48 A_1_M1_M1:54 A_2_M1:54 | N1 N2 N3 N4 N5 N6 |

图10：第8次寻找



- 1.获取F值最小的方块A_2_M1_M1=60，将其加入关闭列表
- 2.寻找A_2_M1_M1周围的可行走的方块，通过与不可行走区域列表和关闭列表进行比较获得
- 3.如果A_2_M1_M1周围的方块存在开放列表中，则计算经A_2_M1_M1到达该方块的G值(从A走，按照父子关系)，如果计算的G值比存在开放列表中的该方块的G值要小的话，则更新开放列表中该方块的G值和F值，且改变该方块的父物体为A_2_M1_M1
- 4.将可行走的方块加入开放列表，F值顺序排列
- 5.将可行走的方块的父物体全部设为A_2_M1_M1
- 6.选取F值最小的，如果F最小，则比价H值，选取H值最小的

此操作更新的开放的列表中的方块有：

A_3_M1_D1:
G值变化:48->40
F值变化:82->74
父物体变化:从A_2_M1变为A_3_M1_M1

开放列表

关闭列表

不可行走区域

| 开放列表 | 关闭列表 | 不可行走区域 |
|--|---|----------------------------------|
| A_1_M1_T1:62 AT3:62 AD1:62 AM2:62 AD2:62 A_1_T2_T3:68 A_2_M1_D1:74 A_2_M1_D1:74 | A:0 AT1:42 A_1_M1:48 AT2:48 AM1:48 A_1_M1_M1:54 A_2_M1:54 | N1 N2 N3 N4 N5 N6 |

图11：最终图



- 1.获取F值最小的方块A_2_M1_M1=60，将其加入关闭列表
- 2.寻找A_2_M1_M1周围的可行走的方块，通过与不可行走区域列表和关闭列表进行比较获得
- 3.如果A_2_M1_M1周围的方块存在开放列表中，则计算经A_2_M1_M1到达该方块的G值(从A走，按照父子关系)，如果计算的G值比存在开放列表中的该方块的G值要小的话，则更新开放列表中该方块的G值和F值，且改变该方块的父物体为A_2_M1_M1
- 4.将可行走的方块加入开放列表，F值顺序排列
- 5.将可行走的方块的父物体全部设为A_2_M1_M1
- 6.选取F值最小的，如果F最小，则比价H值，选取H值最小的

此操作更新的开放的列表中的方块有：

A_3_M1_D1:
G值变化:48->40
F值变化:82->74
父物体变化:从A_2_M1变为A_3_M1_M1

开放列表

关闭列表

不可行走区域

| 开放列表 | 关闭列表 | 不可行走区域 |
|--|---|----------------------------------|
| A_1_M1_T1:62 AT3:62 AD1:62 AM2:62 AD2:62 A_1_T2_T3:68 A_2_M1_D1:74 A_2_M1_D1:74 | A:0 AT1:42 A_1_M1:48 AT2:48 AM1:48 A_1_M1_M1:54 A_2_M1:54 | N1 N2 N3 N4 N5 N6 |