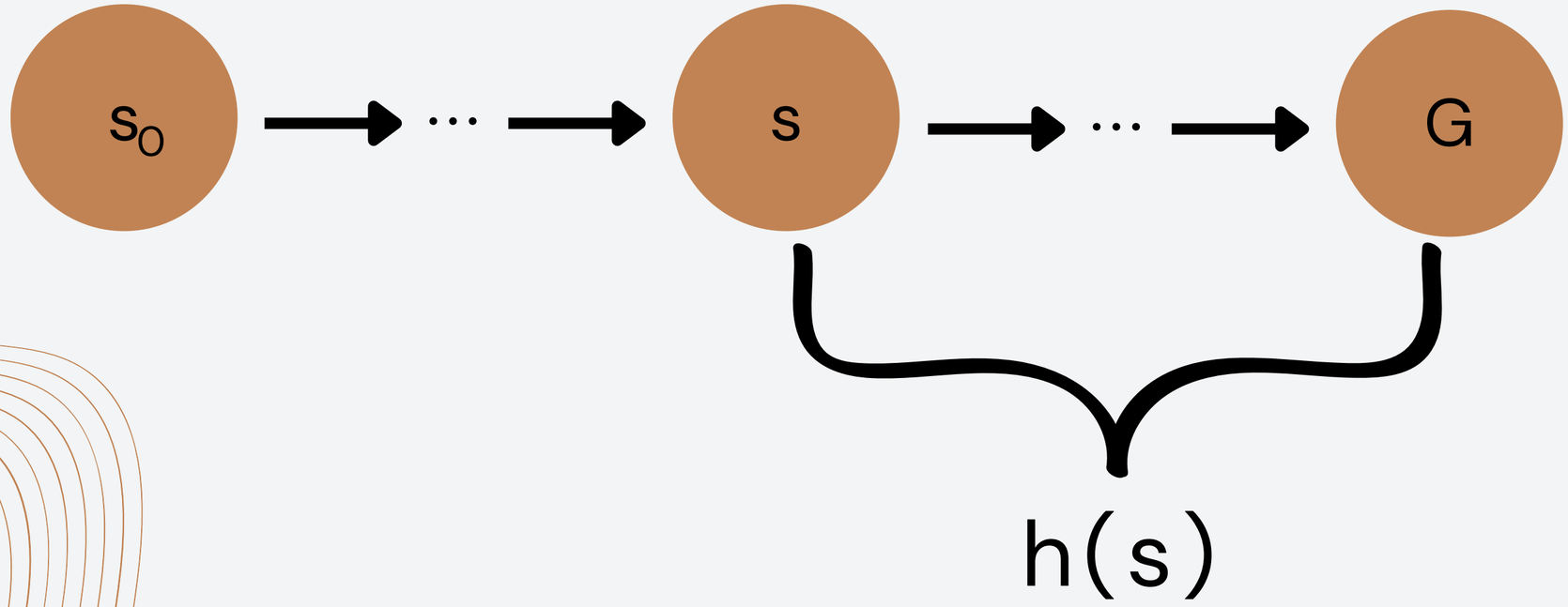
AYUDANTÍA 05

A*

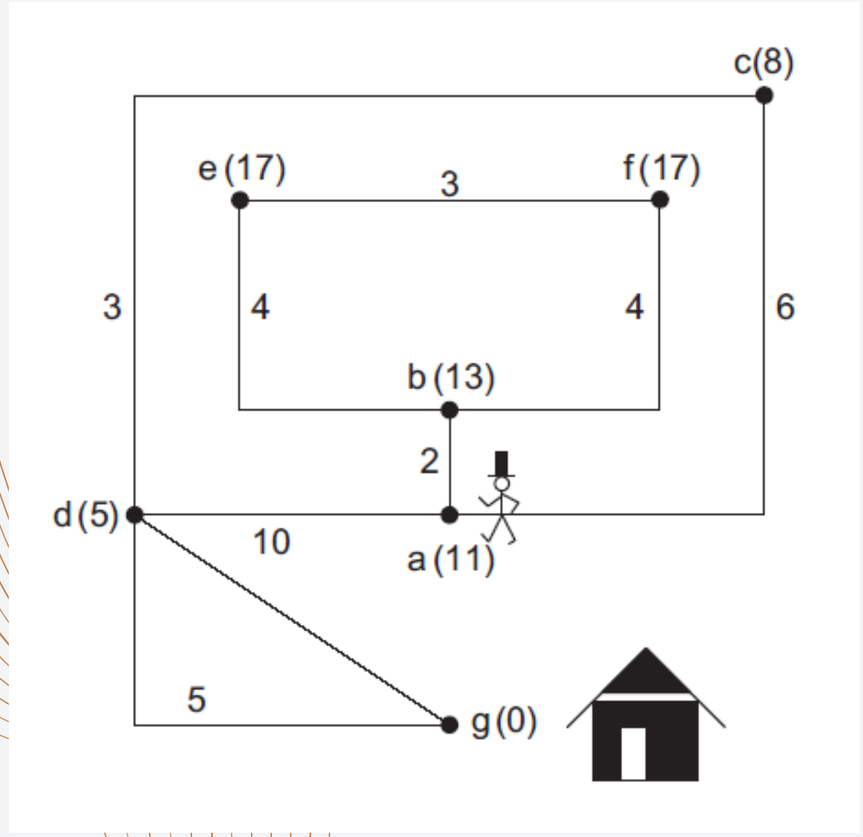
HEURÍSTICAS

HEURÍSTICAS

Es una función que estima qué tan cerca está un estado s del estado final G

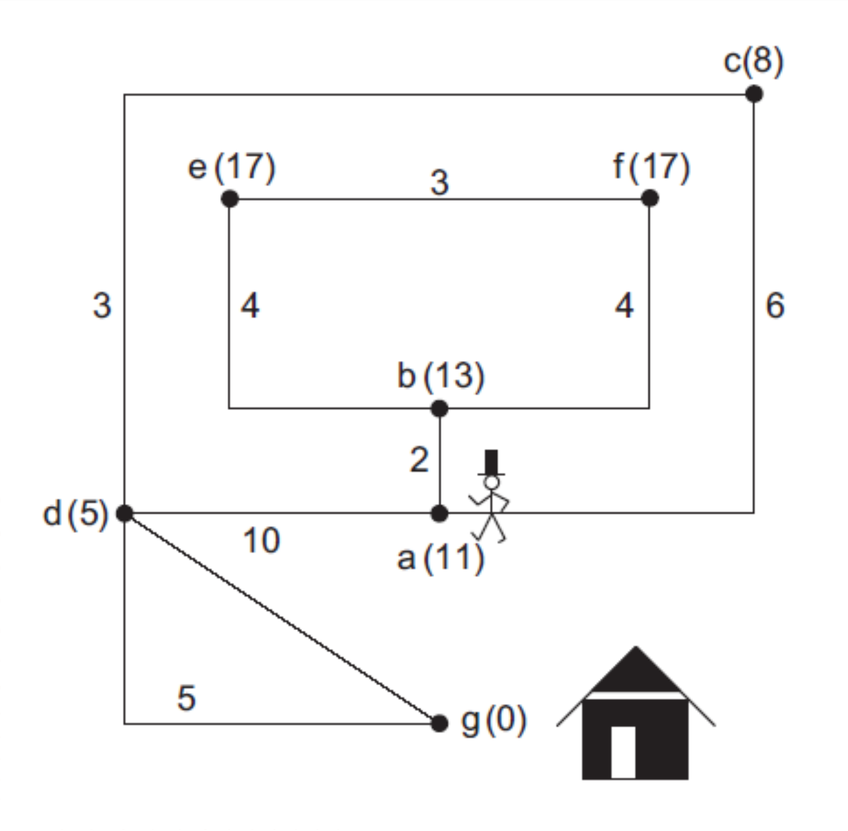


HEURÍSTICAS



s=	a	b	c	d	e	f	g
h(s)=	11	13	8	5	17	17	0

HEURÍSTICAS ADMISIBLES



s=	a	b	c	d	e	f	g
h(s)=	11	13	8	5	17	17	0
h*(s)=	14	16	8	5	20	20	0

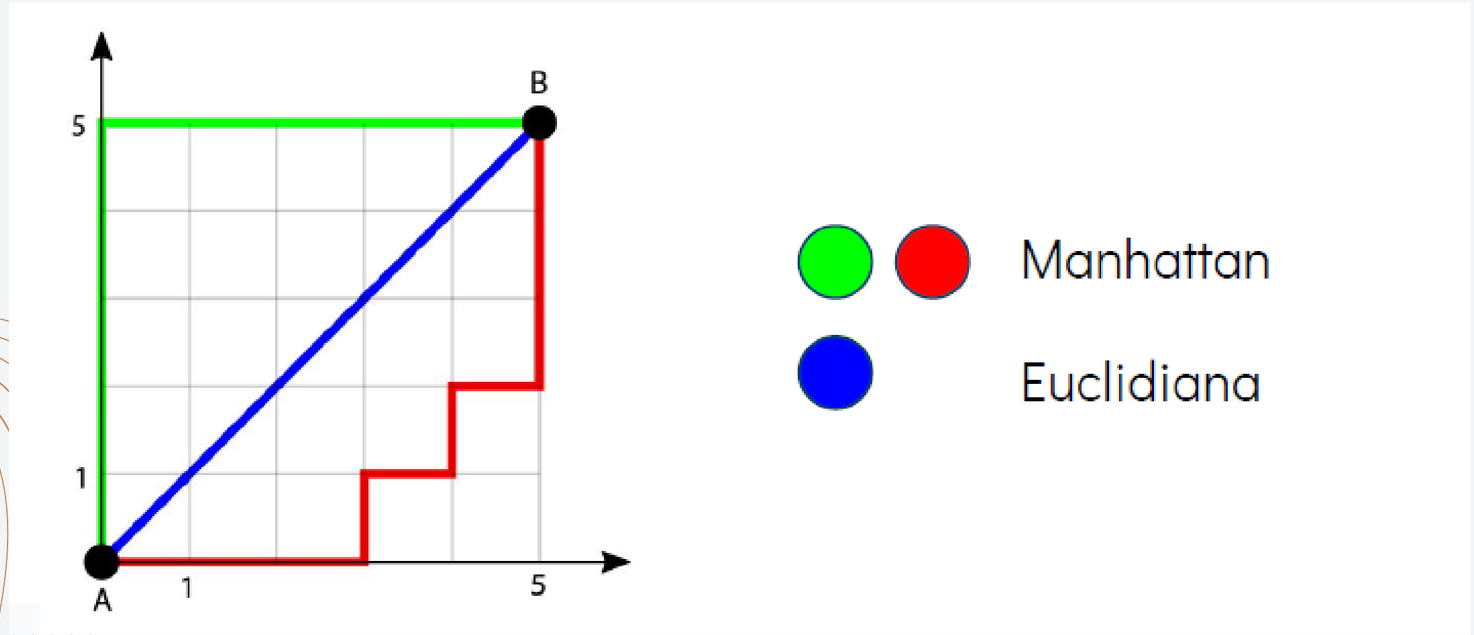
h*(s): Costo del camino óptimo desde s

DISTANCIA MANHATTAN Y EUCLIDIANA

La distancia entre dos puntos es:

- Manhattan: la suma de las diferencias absolutas de sus coordenadas. $D_{Manhattan} = |x_1 - x_2| + |y_1 - y_2|$
- Euclidiana: la distancia en línea recta entre los puntos

$$D_{Euclidiana} = \sqrt{[(x_1 - x_2)^2 + (y_1 - y_2)^2]}$$



ALGORITMO A*

ALGORITMO A*

Algoritmo A*
Input: Un problema de búsqueda $(\mathbf{S}, A, s_0, G)$
Output: Un nodo objetivo

1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
8. $Insertar\ v$

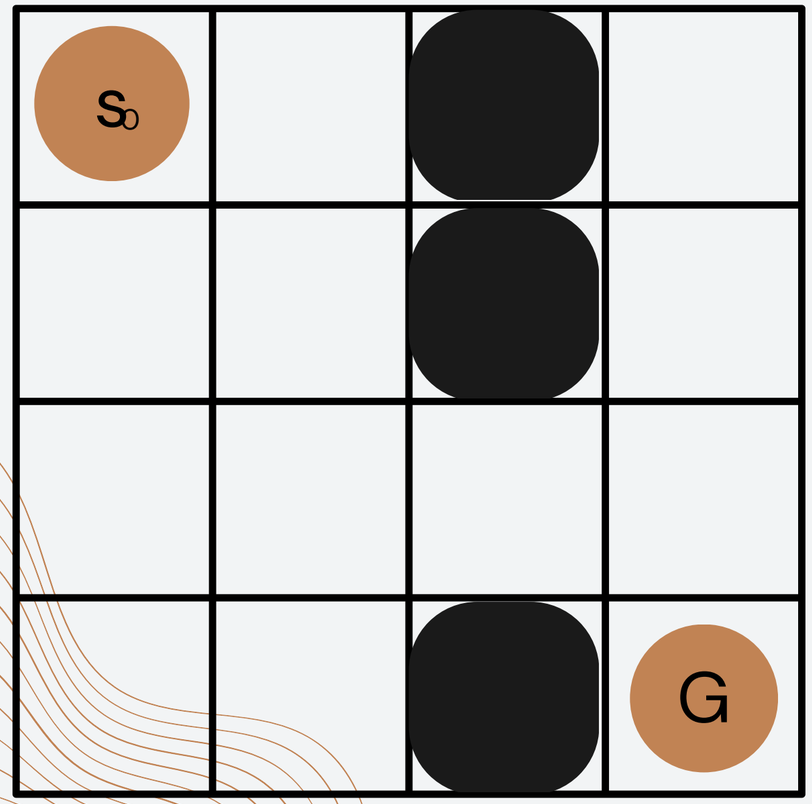
Open se ordena según menor valor de *f*

f(s) = g(s)+h(s)

Se revisa si se llega al nodo objetivo cuando se 'expande'

¿s=G?

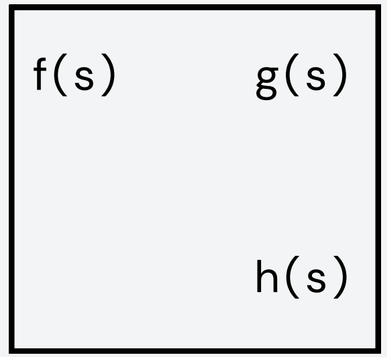
EJEMPLO



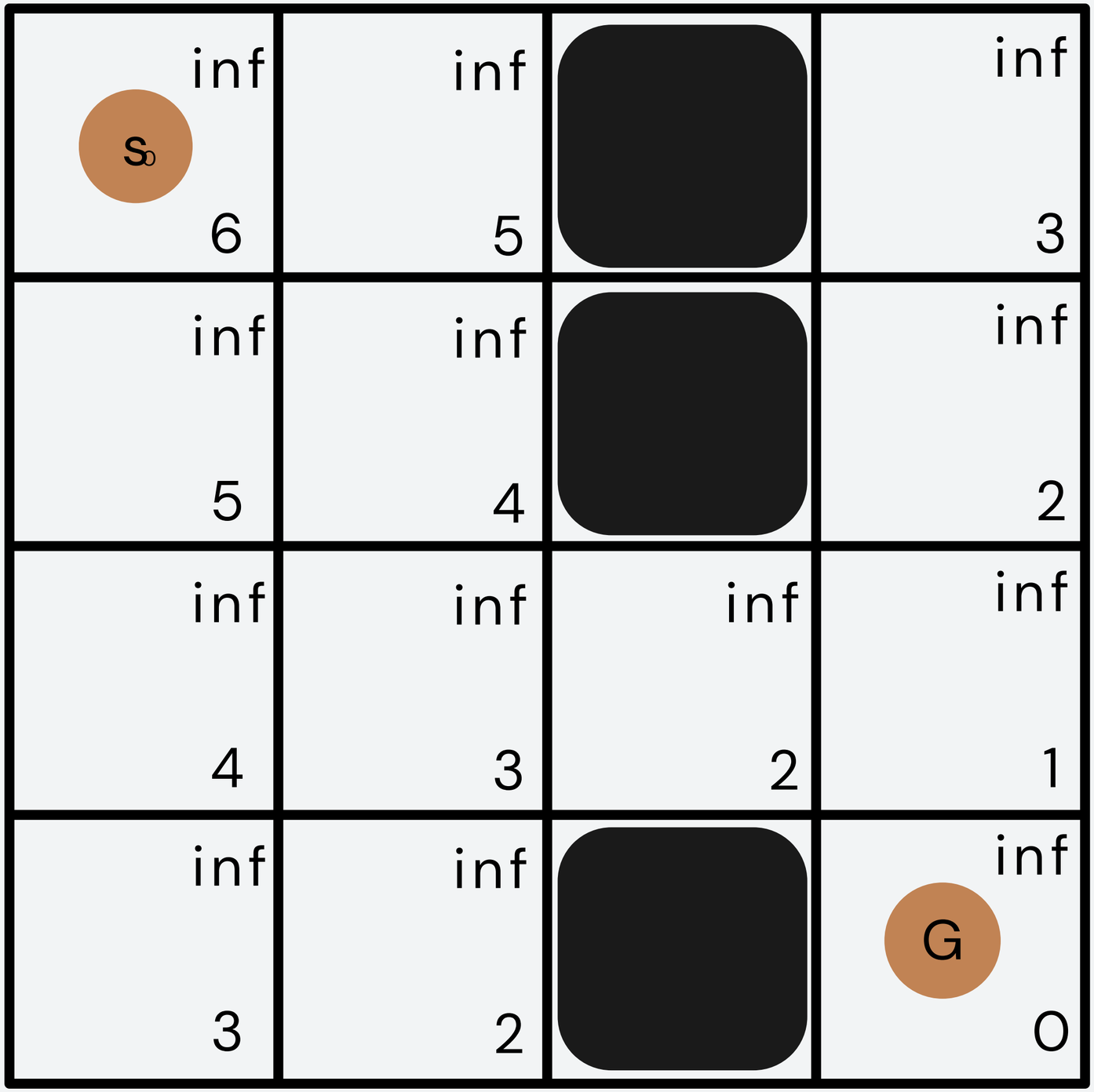
$$f(s) = g(s)+h(s)$$

g(s): largo del camino

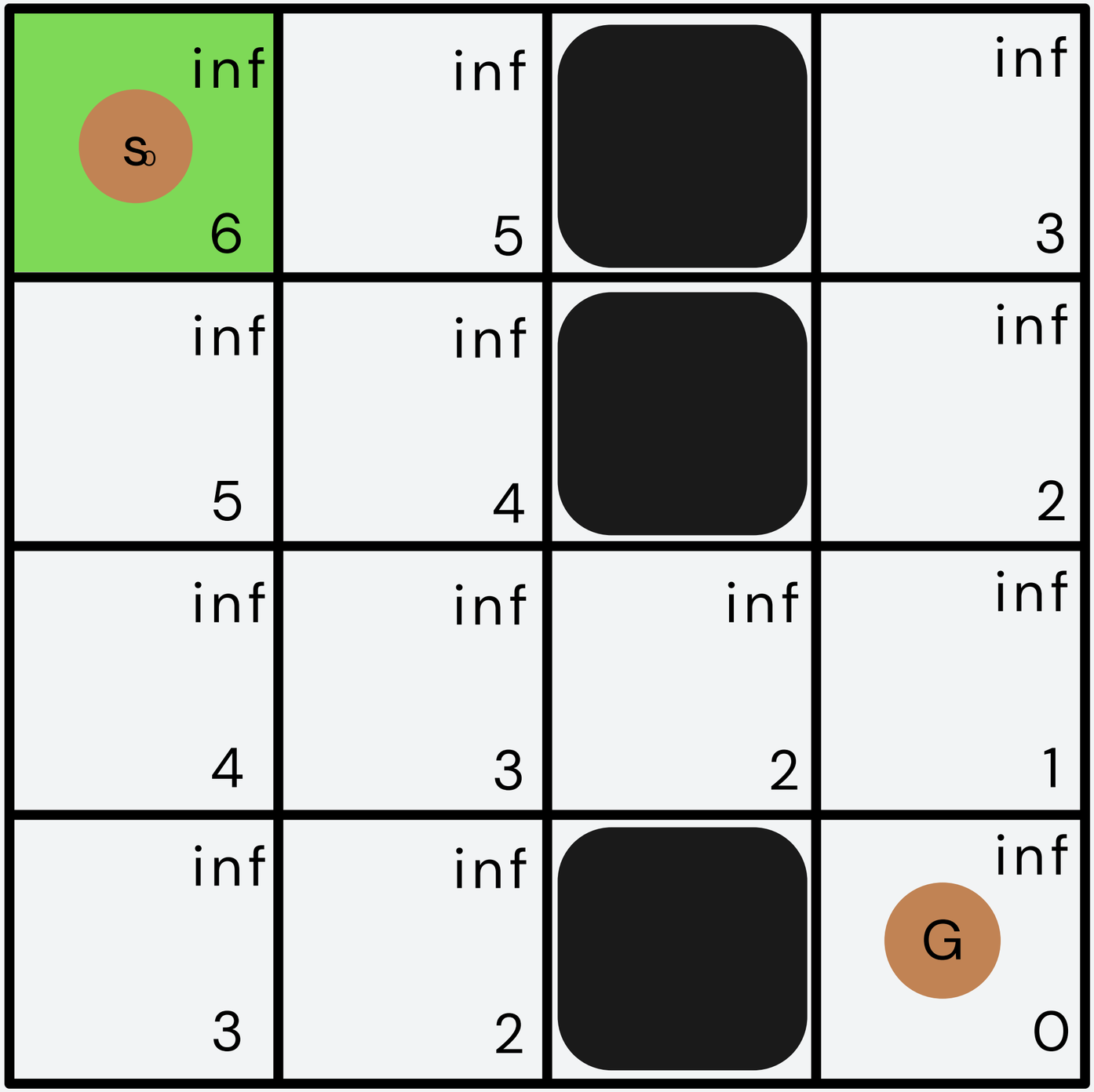
h(s): distancia Manhattan



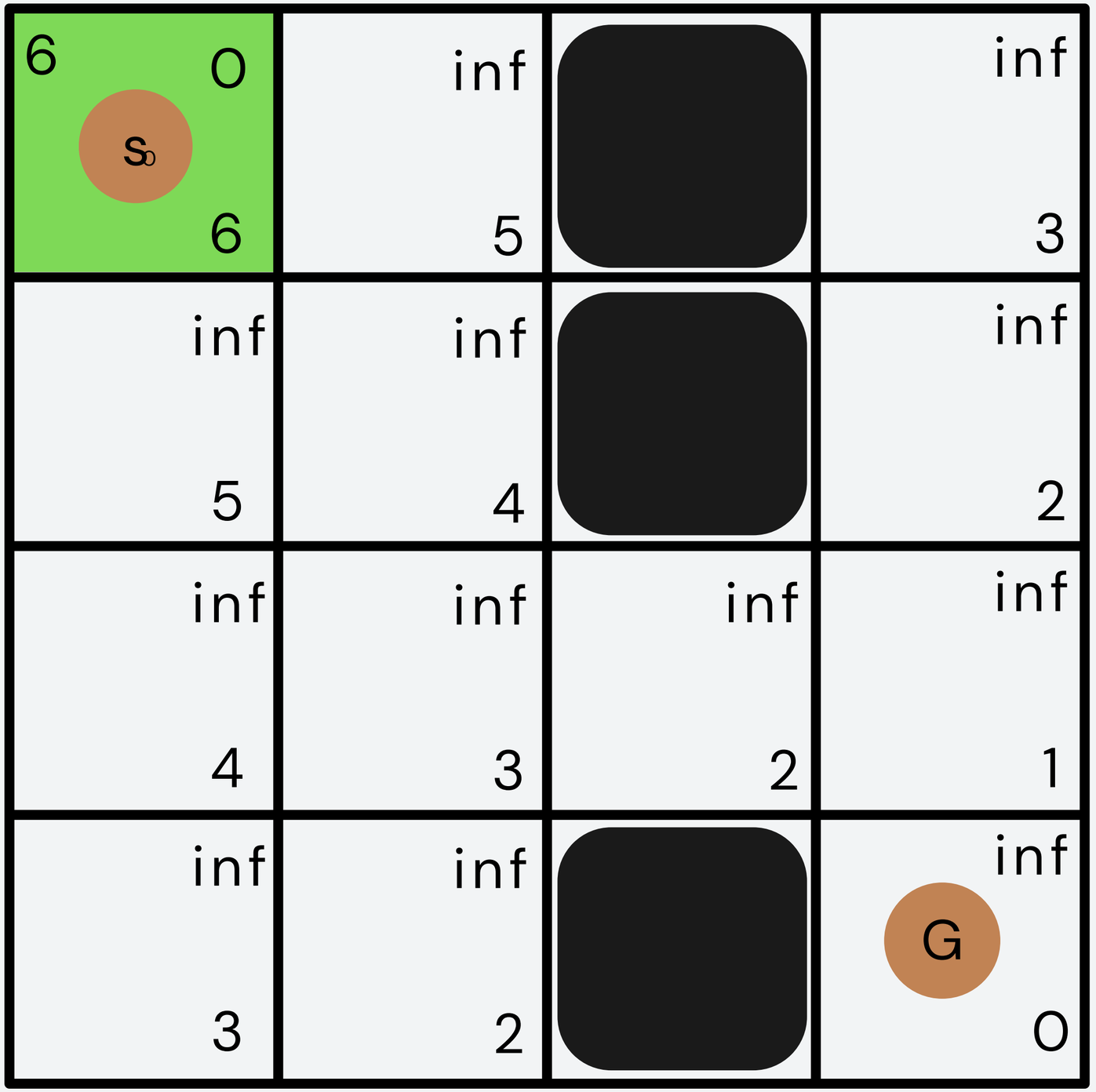
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



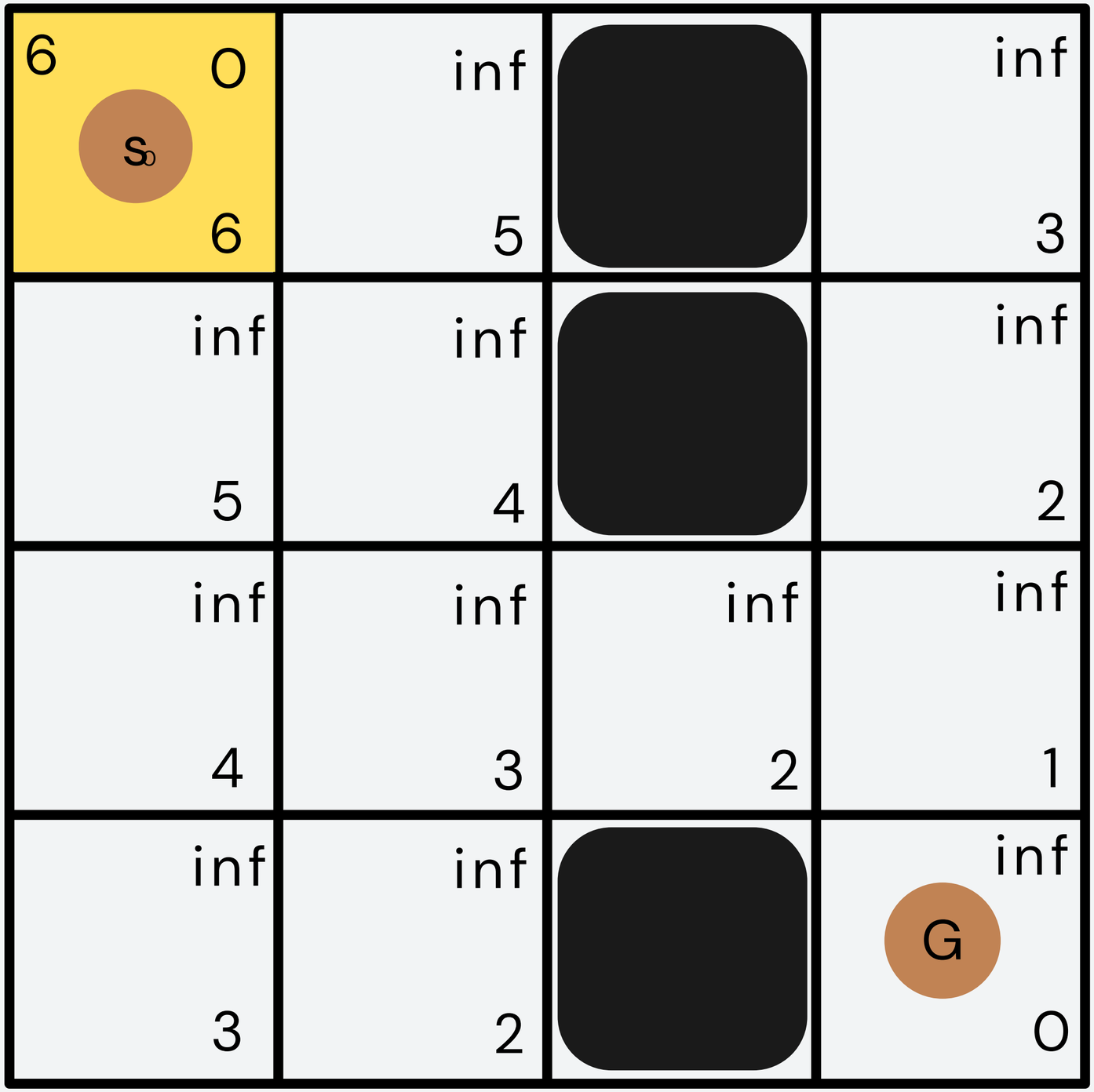
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



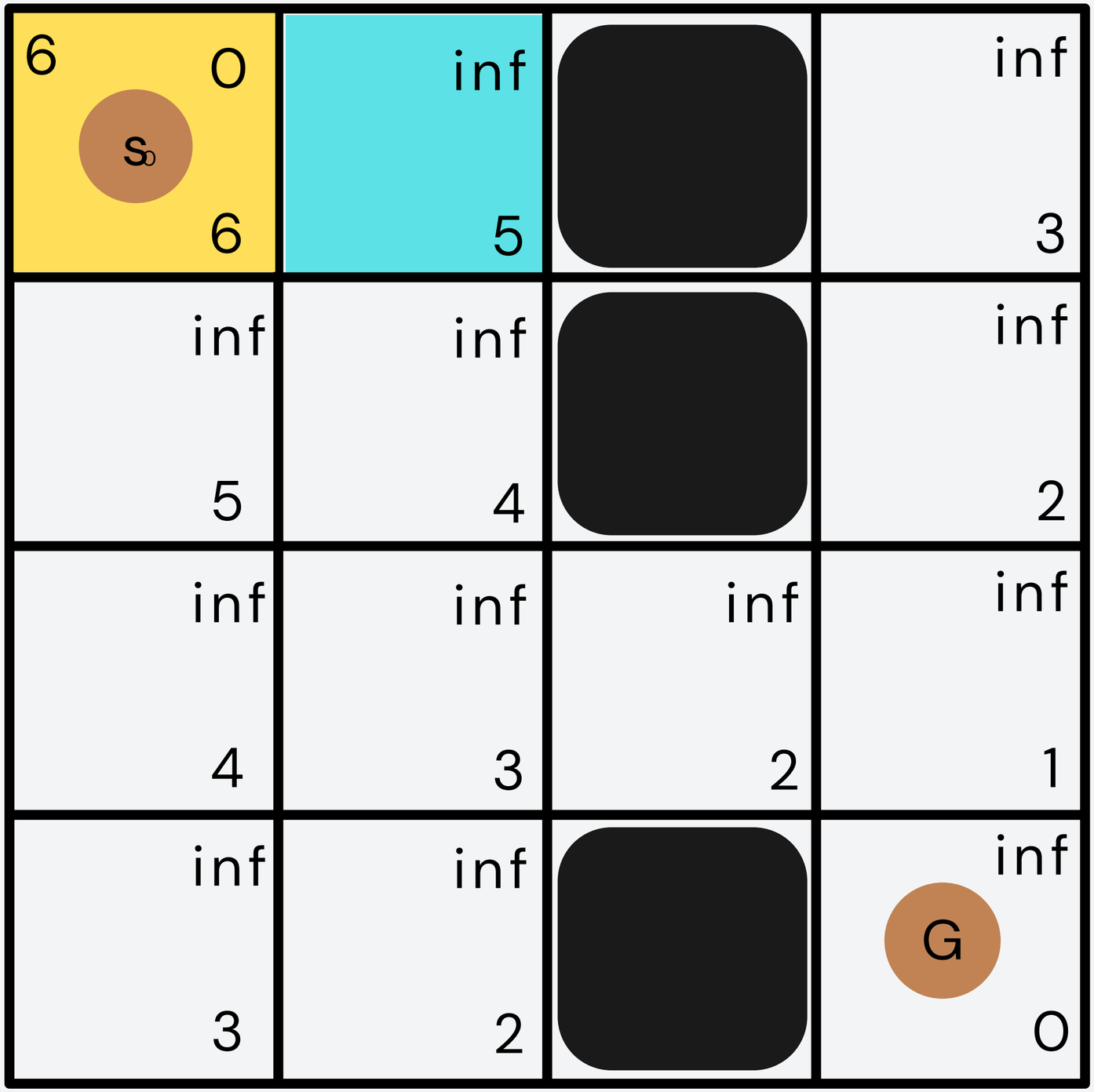
1 **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2 $Open \leftarrow \{s_0\}$
3 $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4 **while** $Open \neq \emptyset$
5 Extrae un u desde $Open$ con menor valor-f
6 **if** u es objetivo **return** u
7 **for each** $v \in Succ(u)$ **do**
1 $cost_v = g(u) + c(u, v)$
2 **if** $cost_v \geq g(v)$ **return**
3 $parent(v) \leftarrow u$
4 $g(v) \leftarrow cost_v$
5 $f(v) \leftarrow g(v) + h(v)$
6 **if** $v \in Open$ **then** Reordenar $Open$
7 **else** Insertar v en $Open$



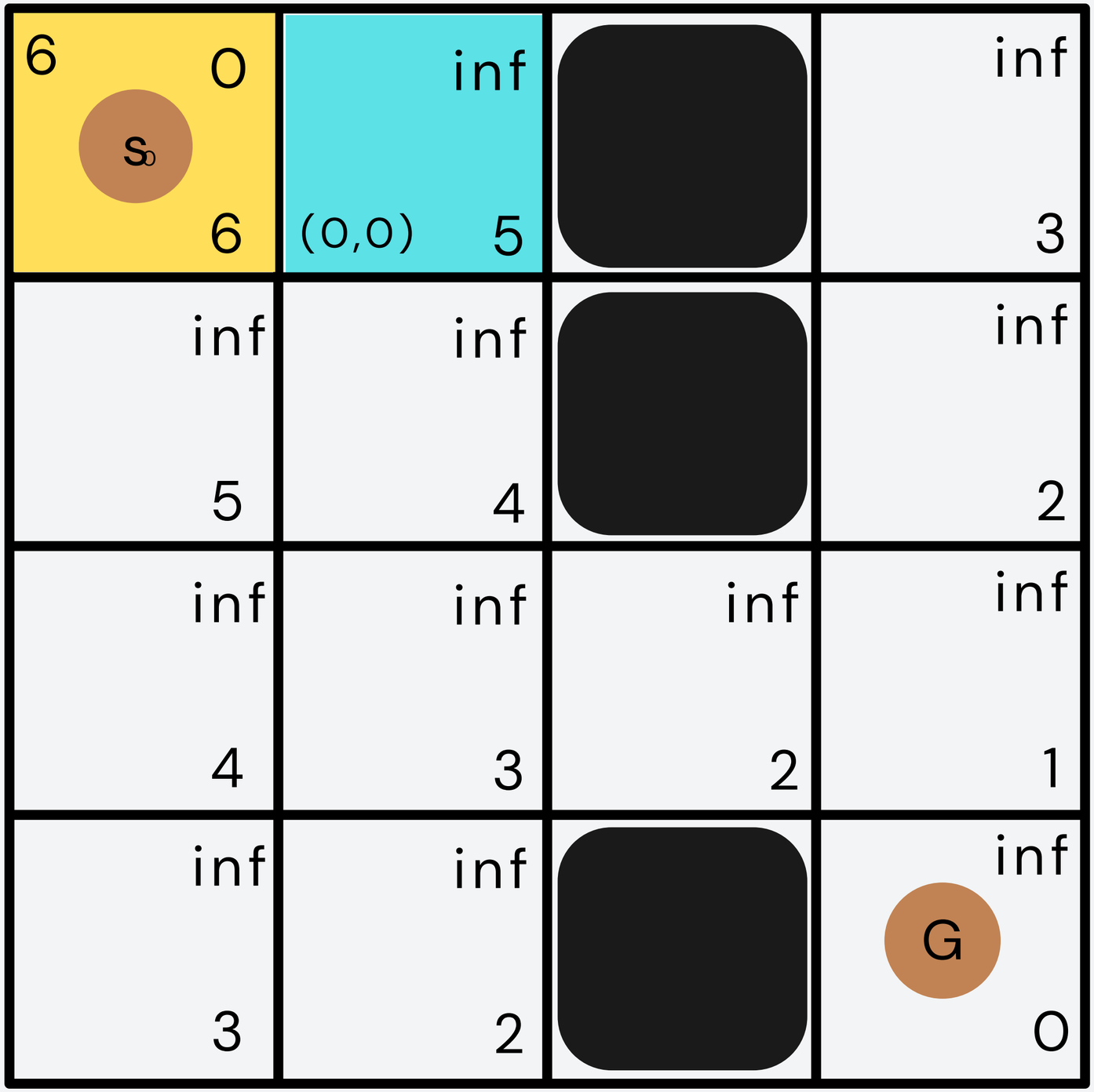
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



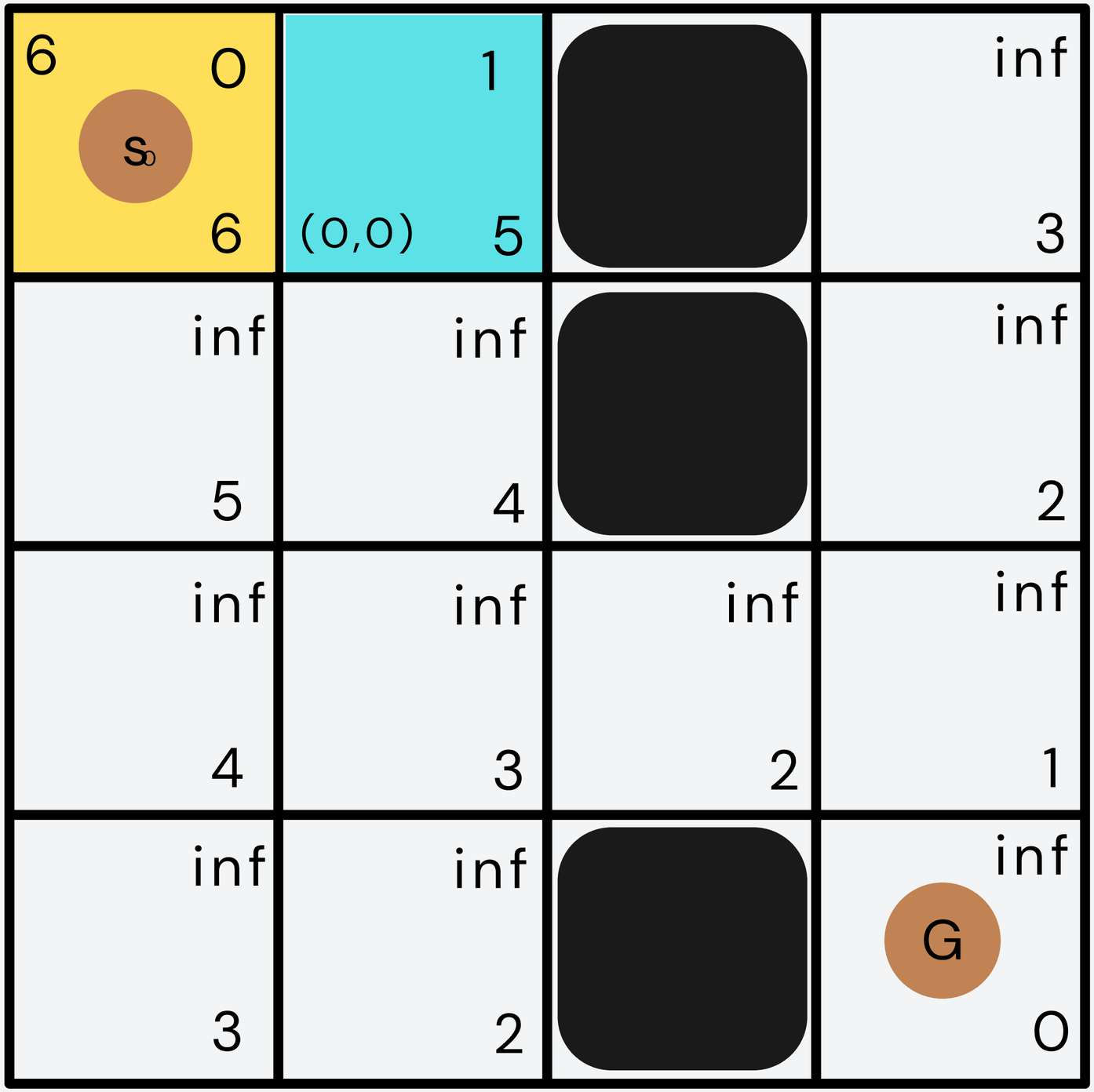
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



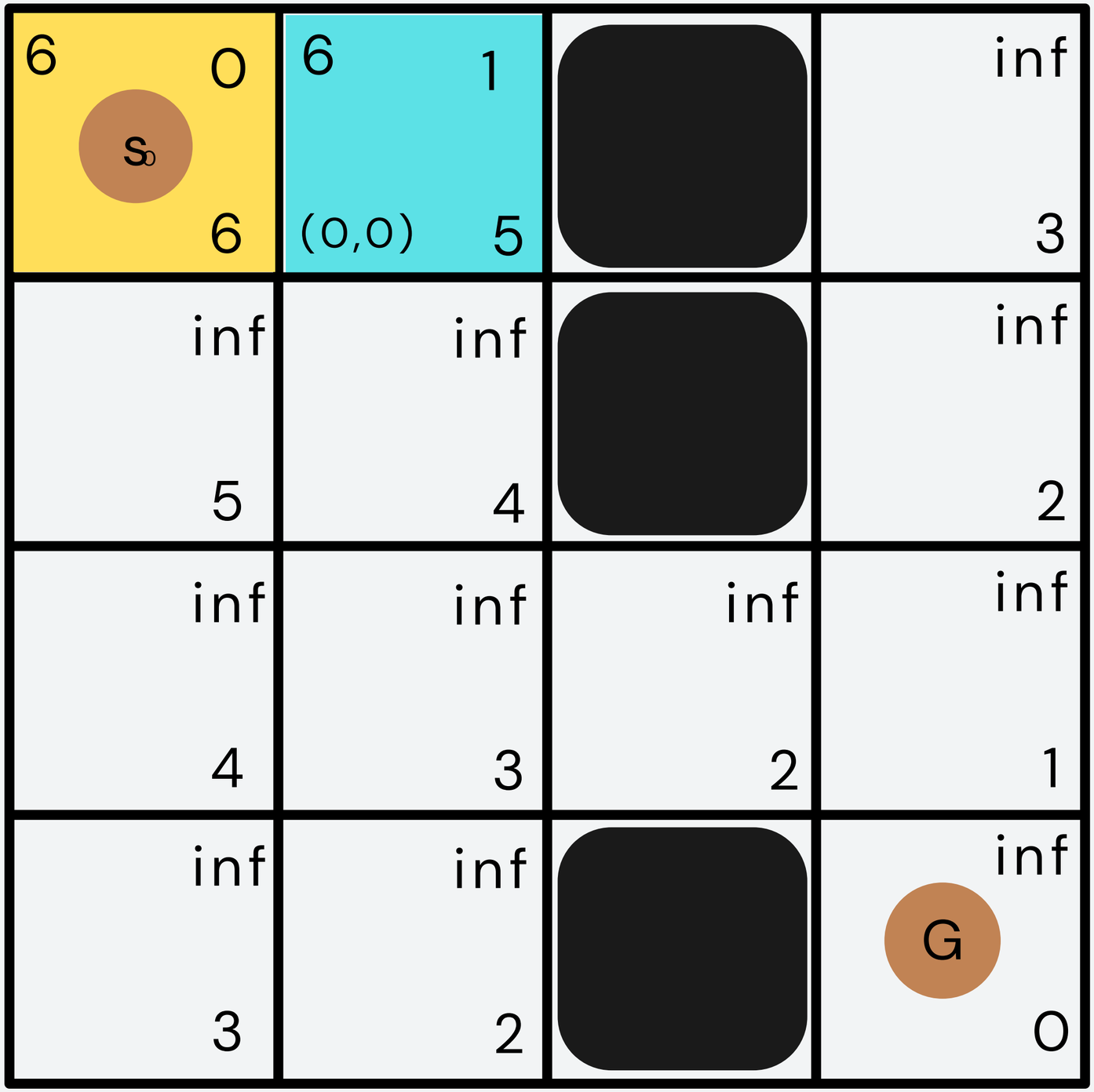
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



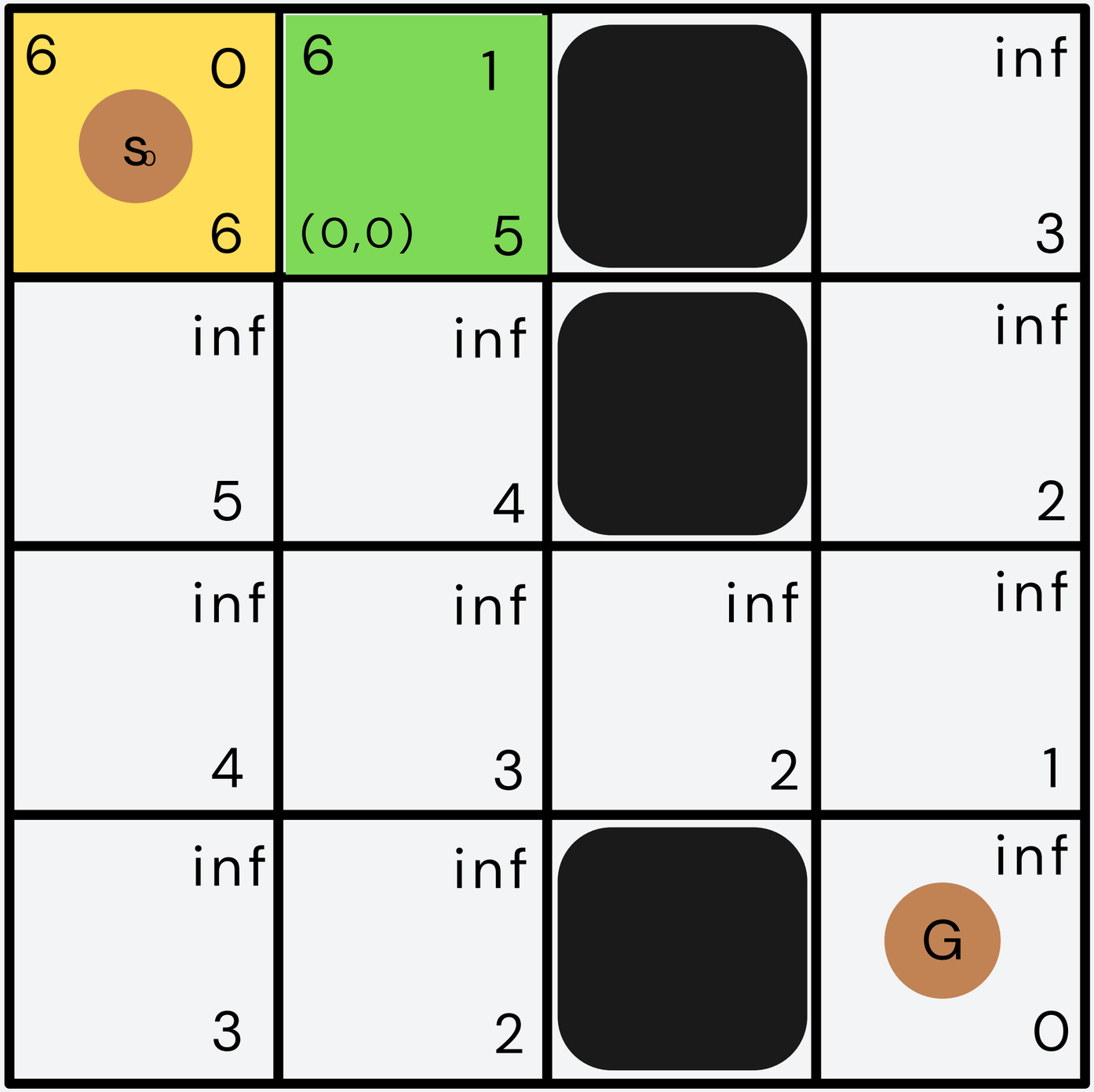
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



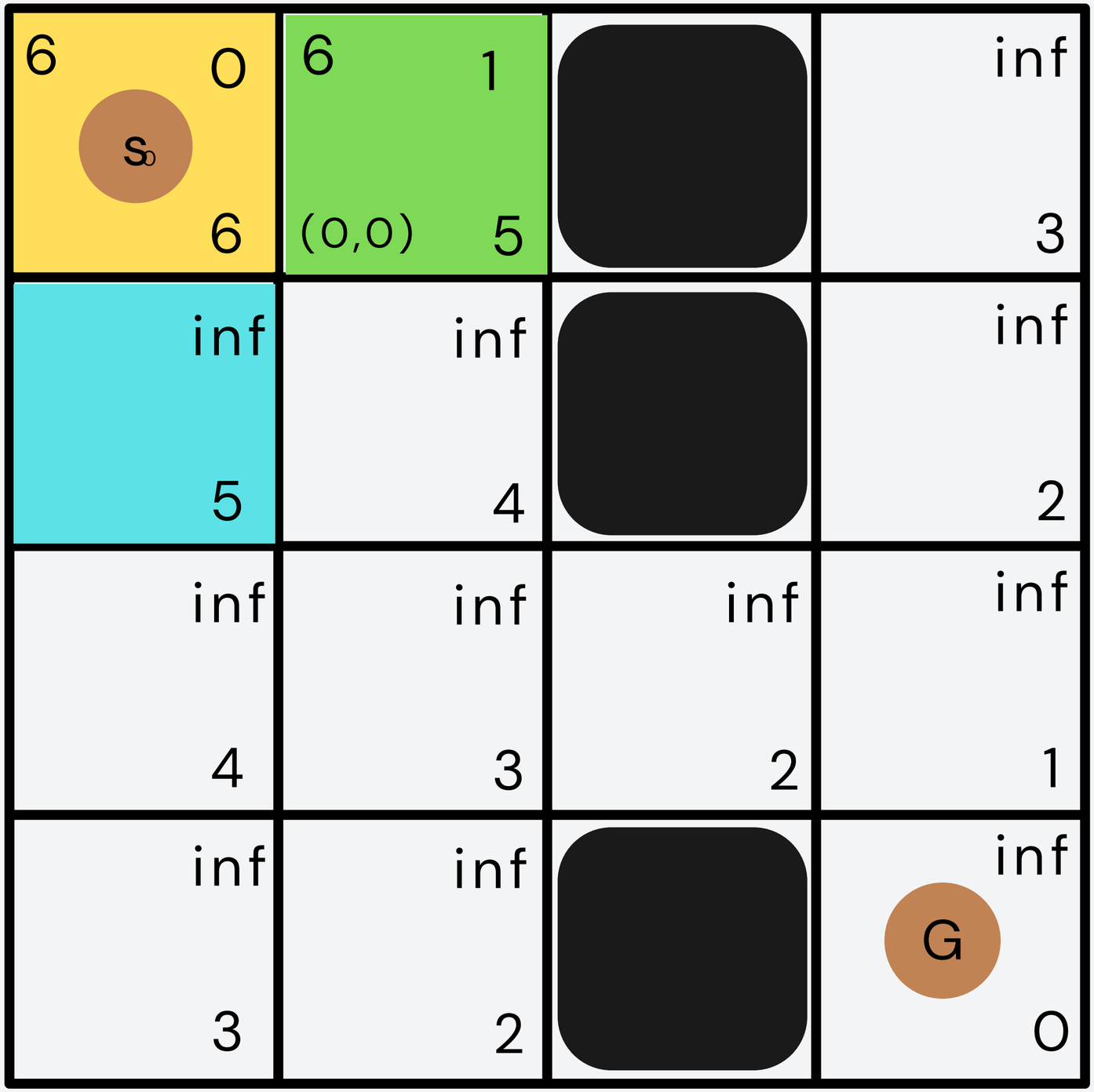
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



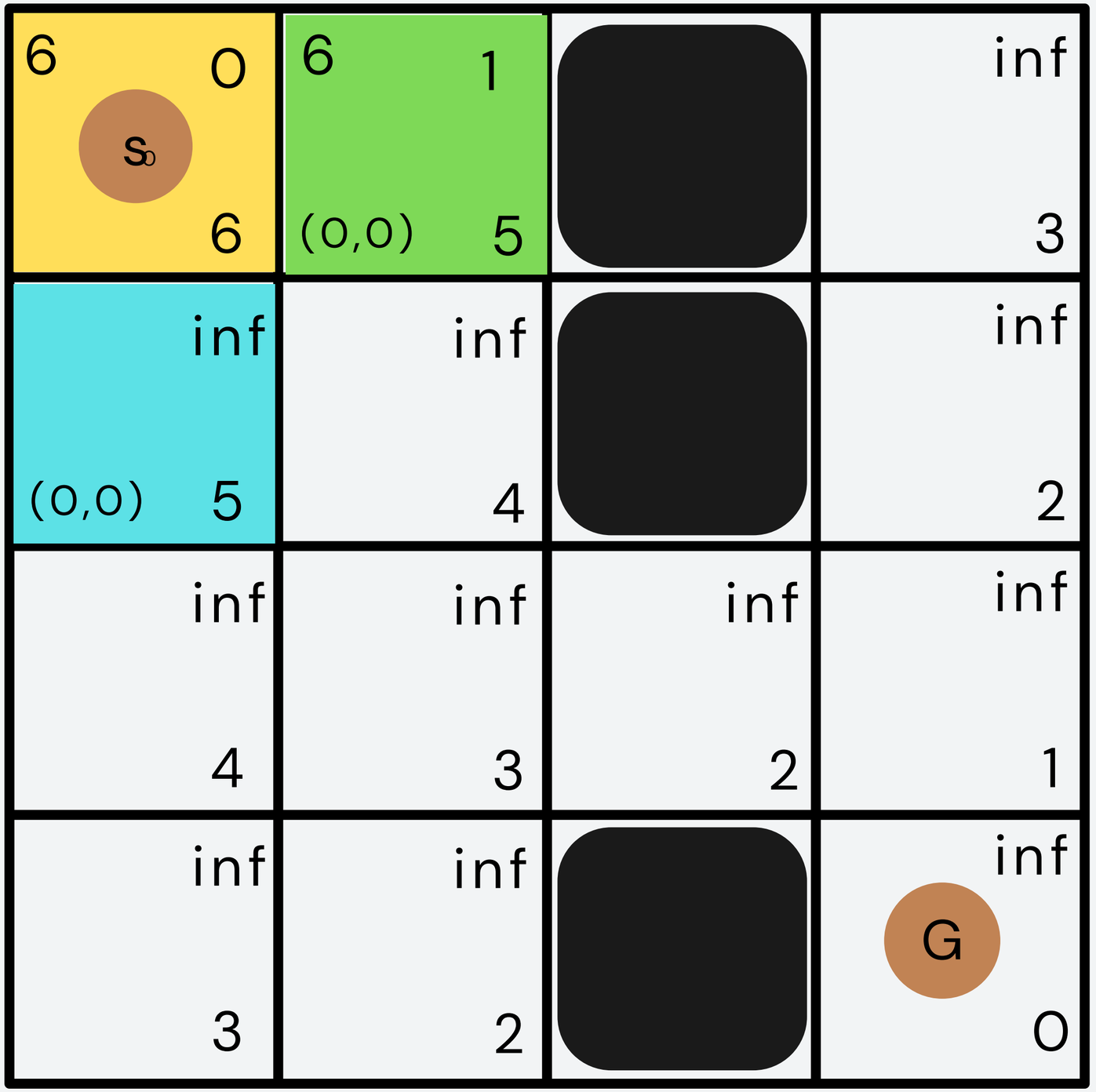
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



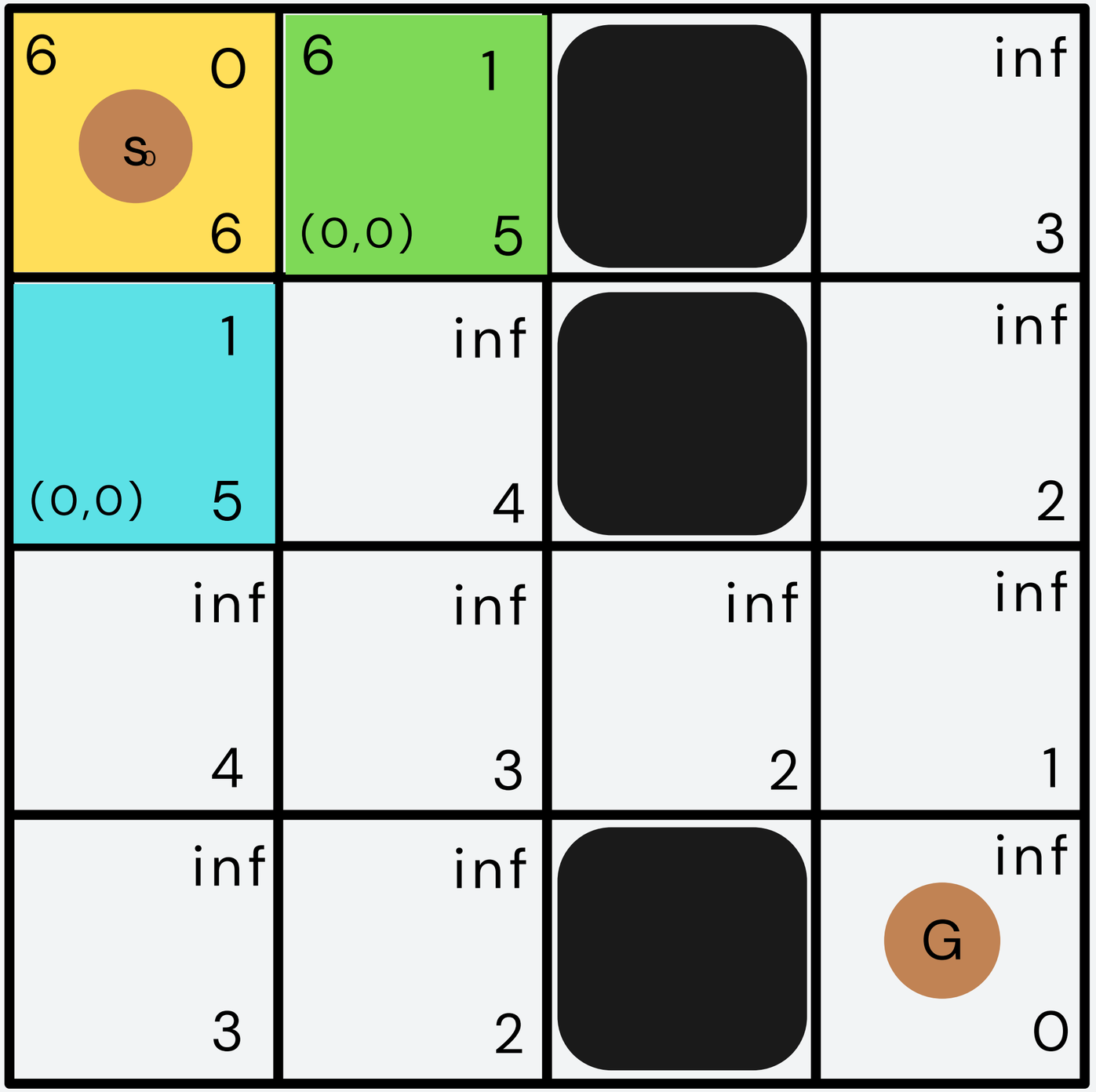
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



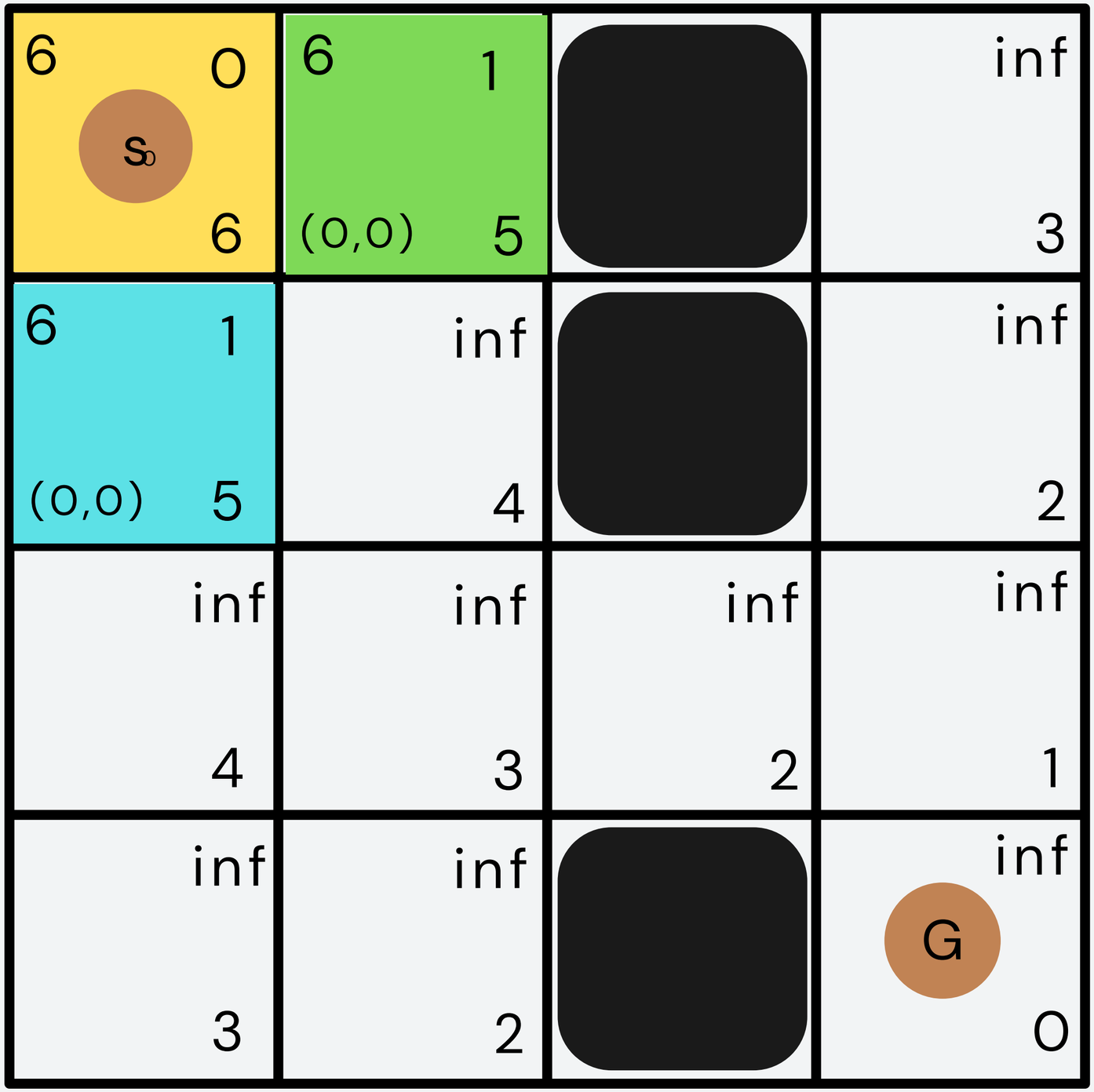
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



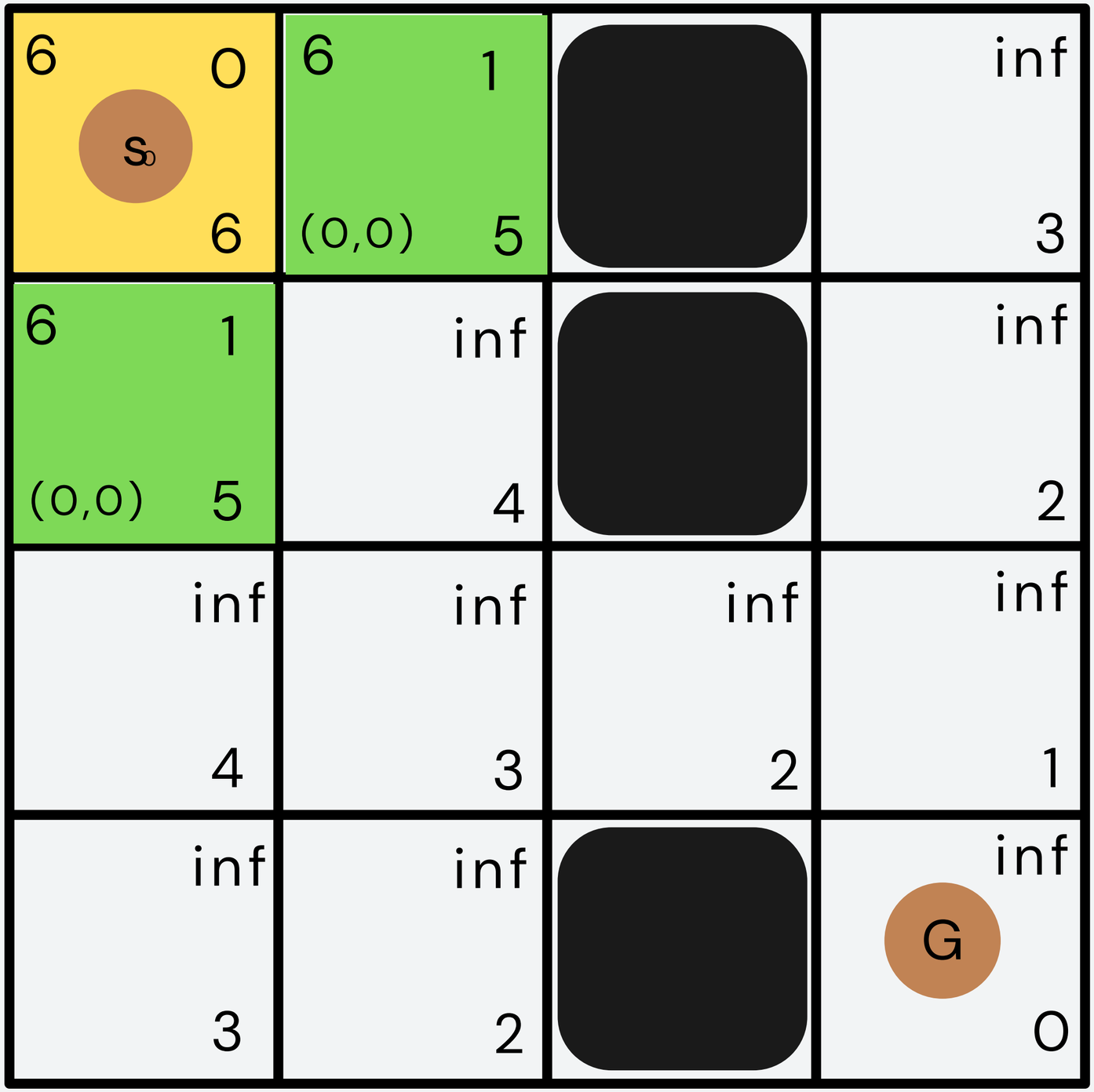
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



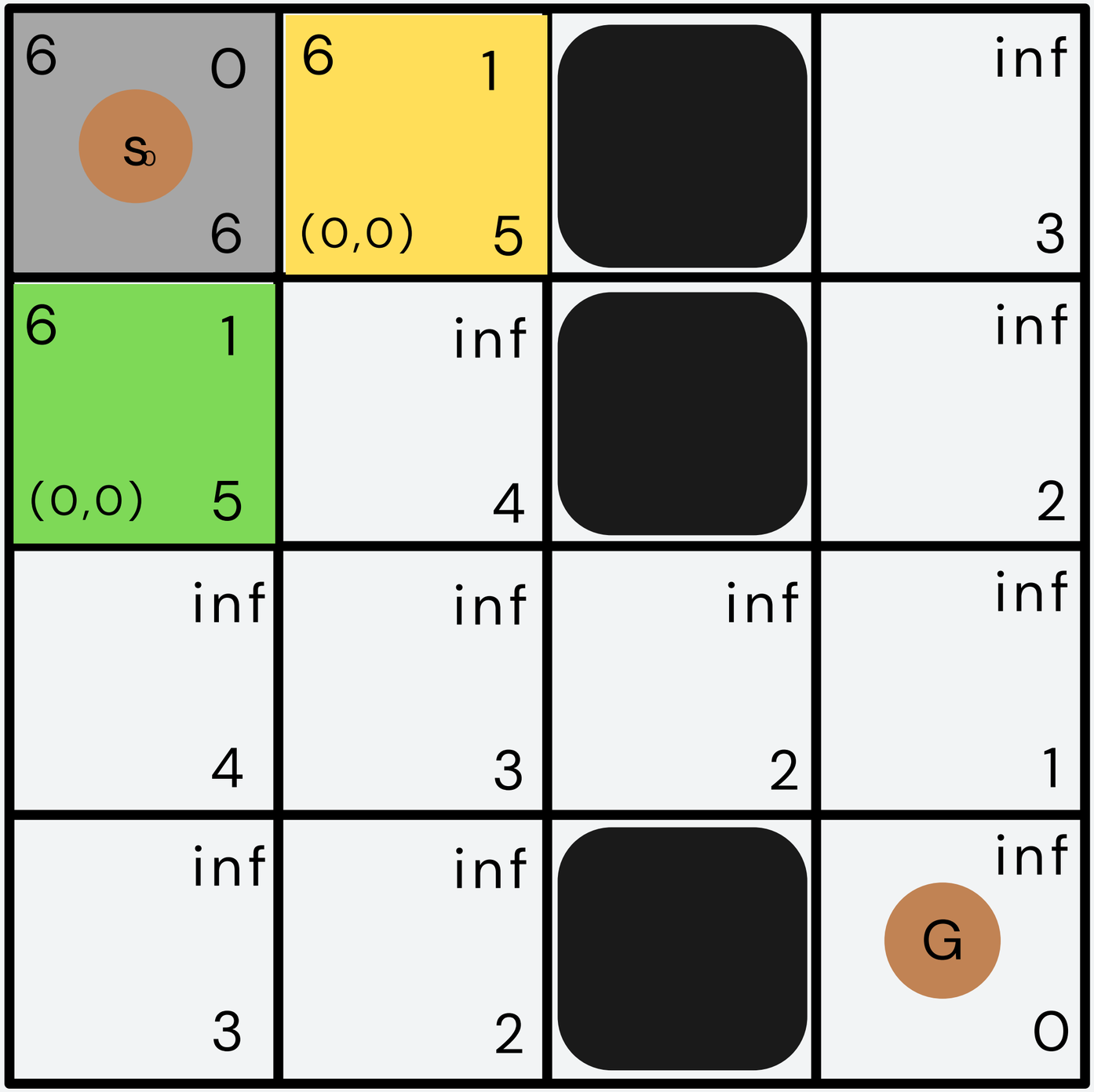
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



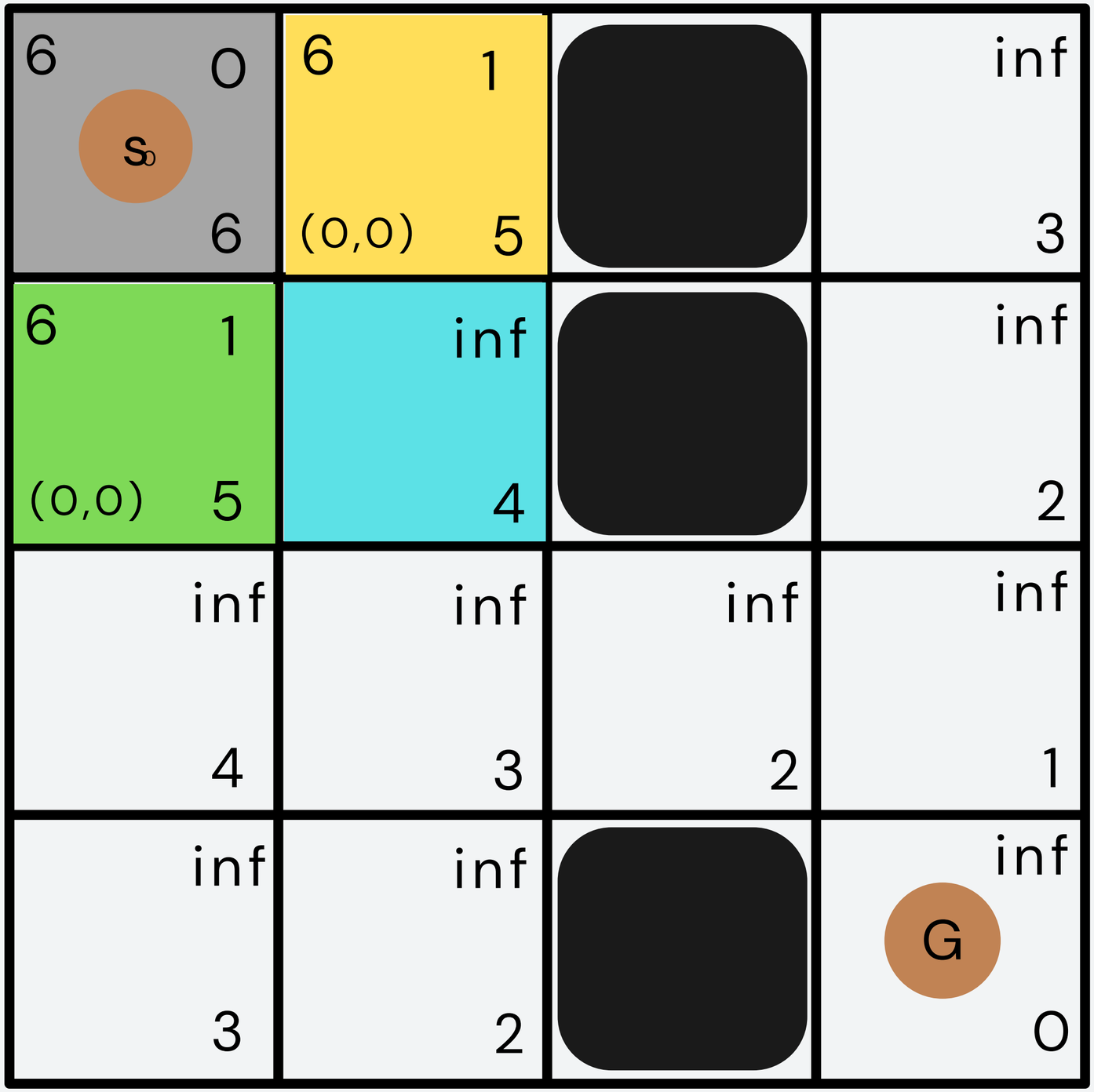
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



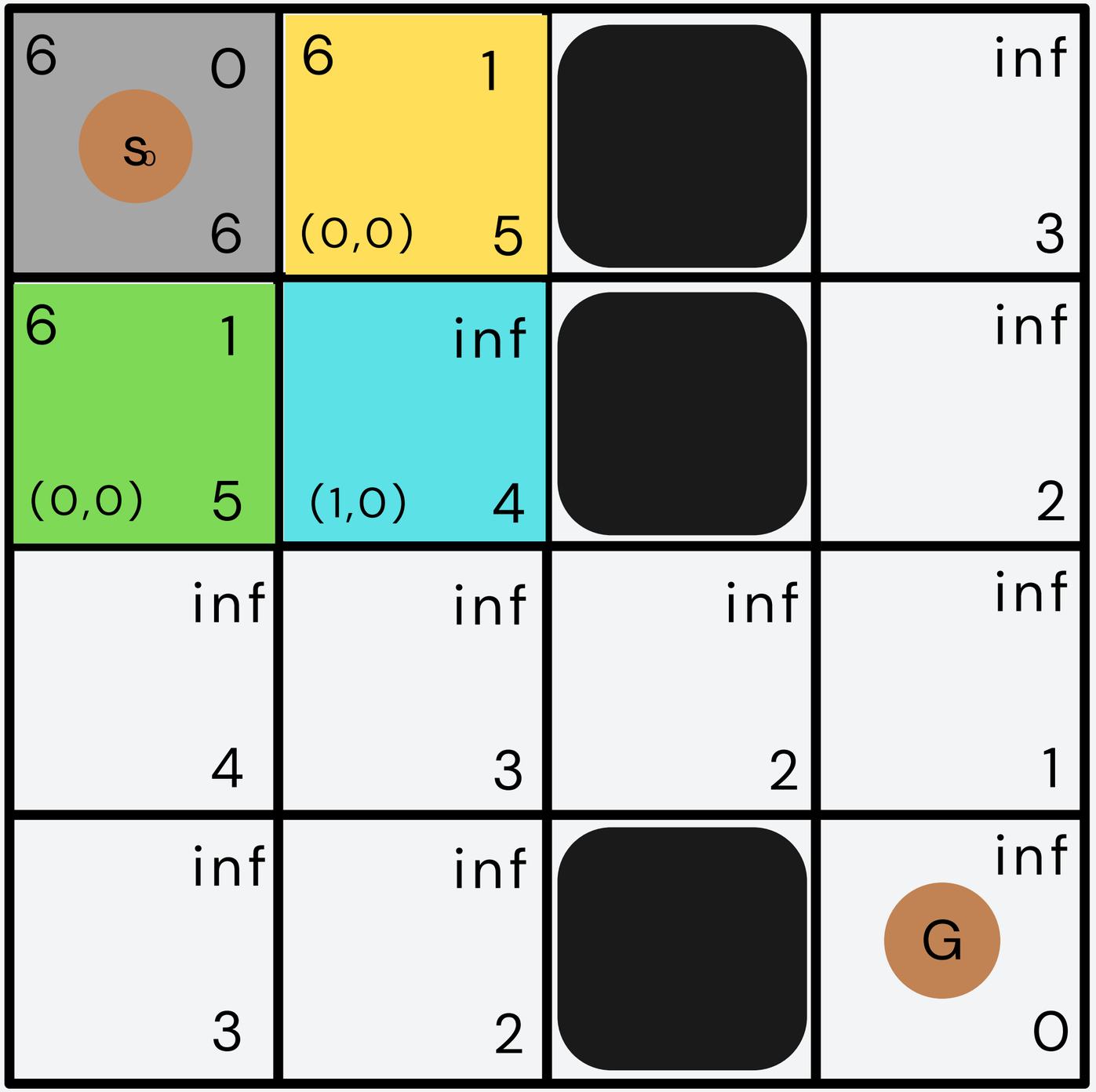
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



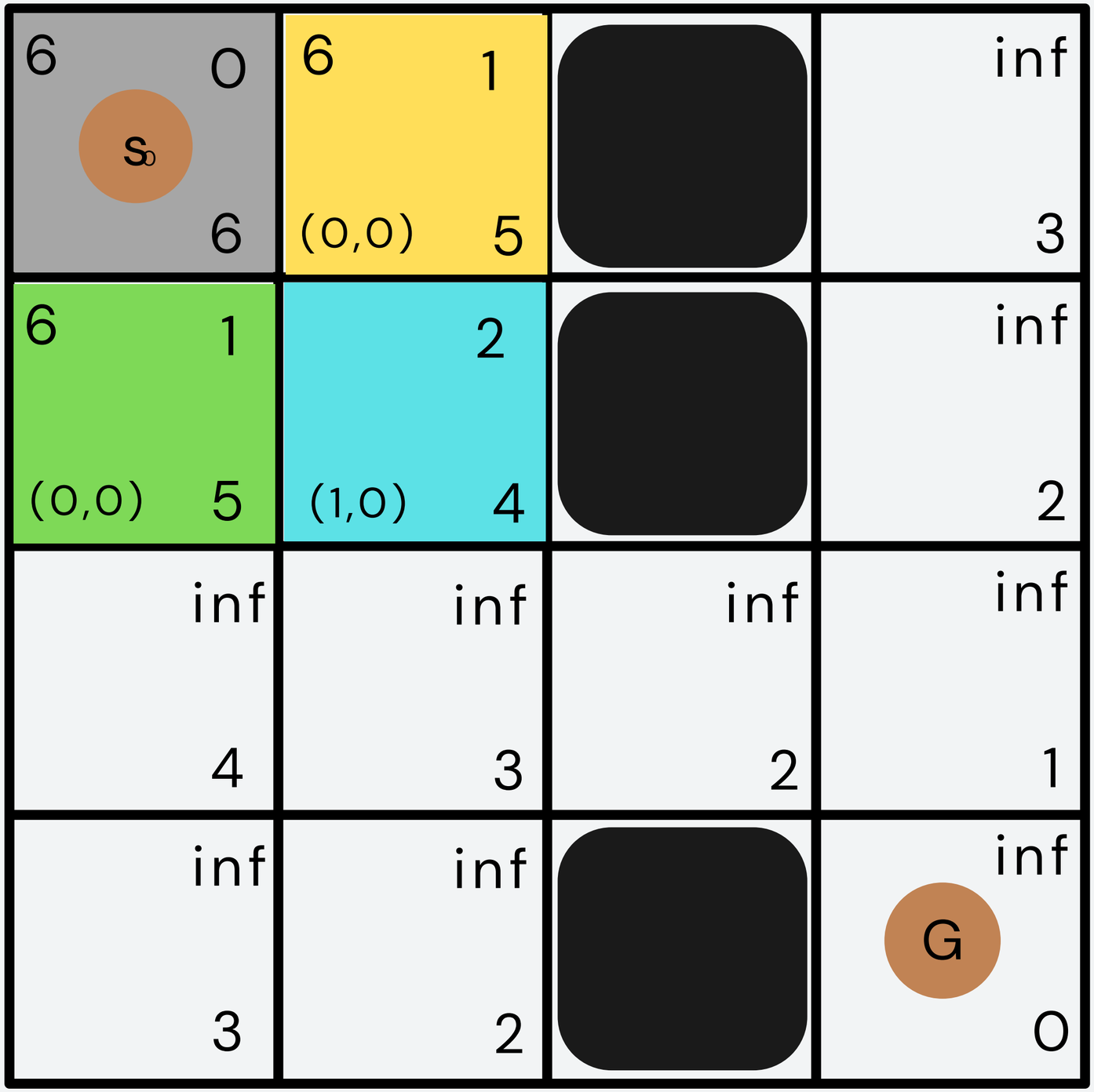
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



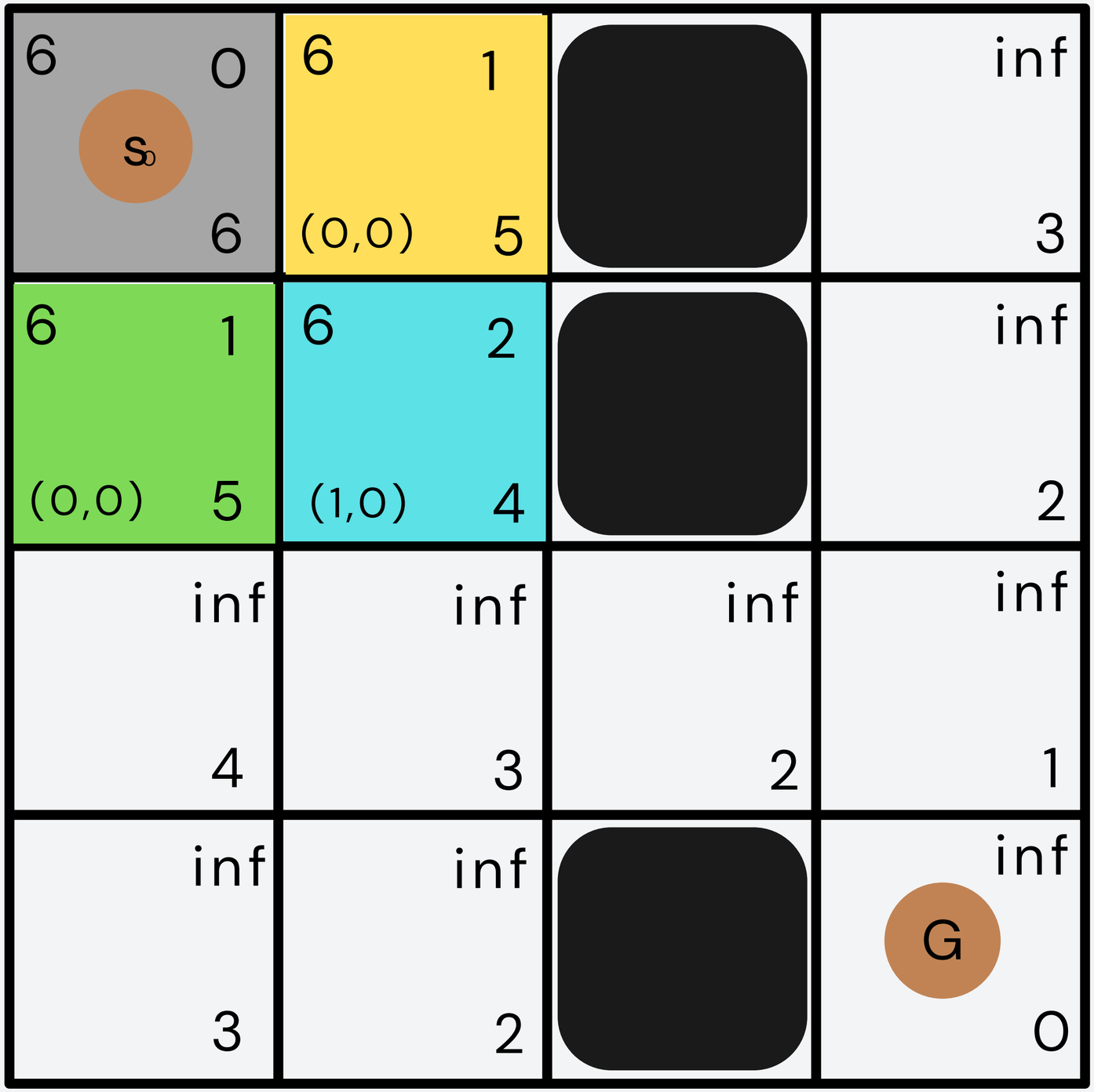
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



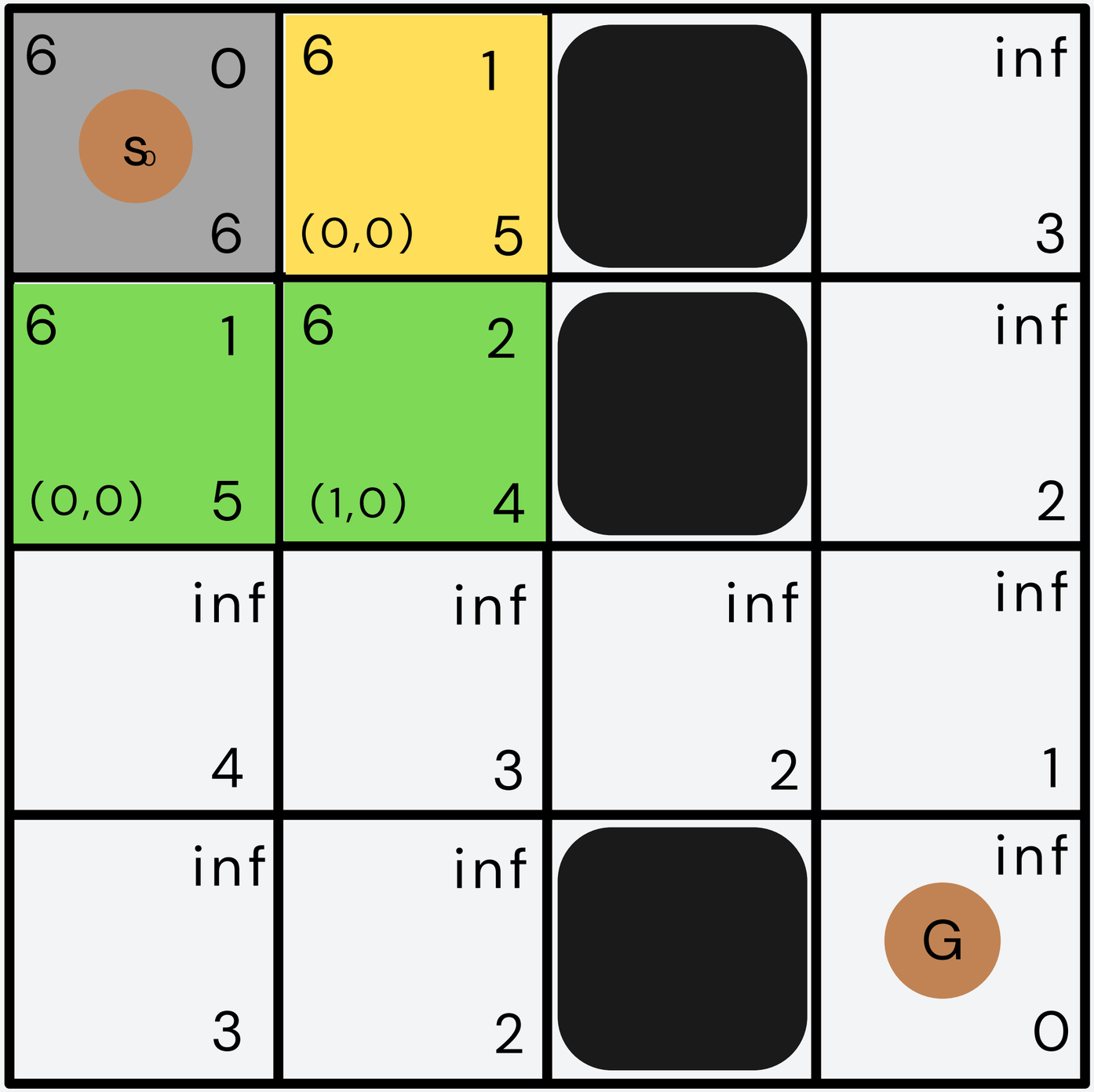
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



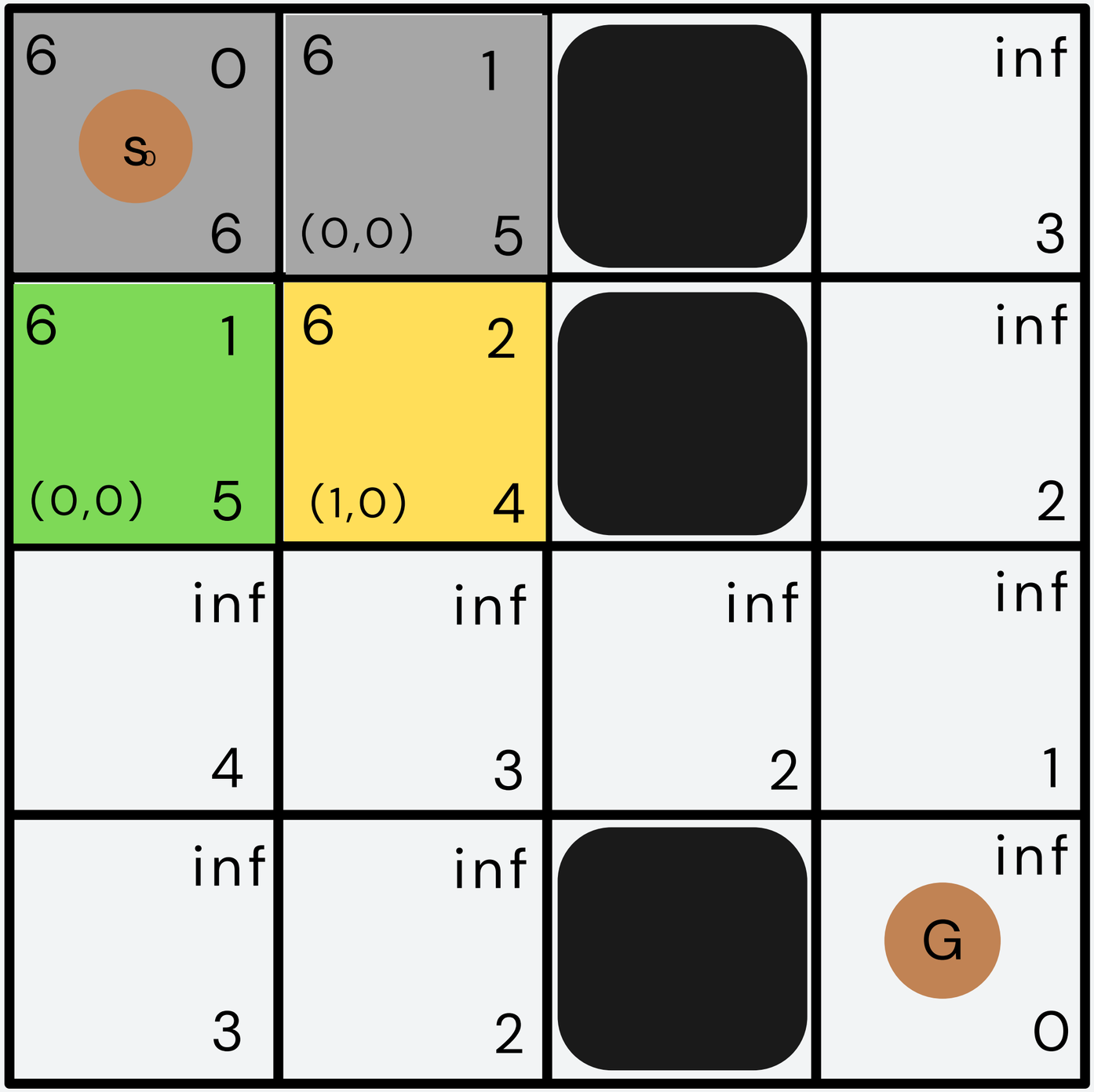
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



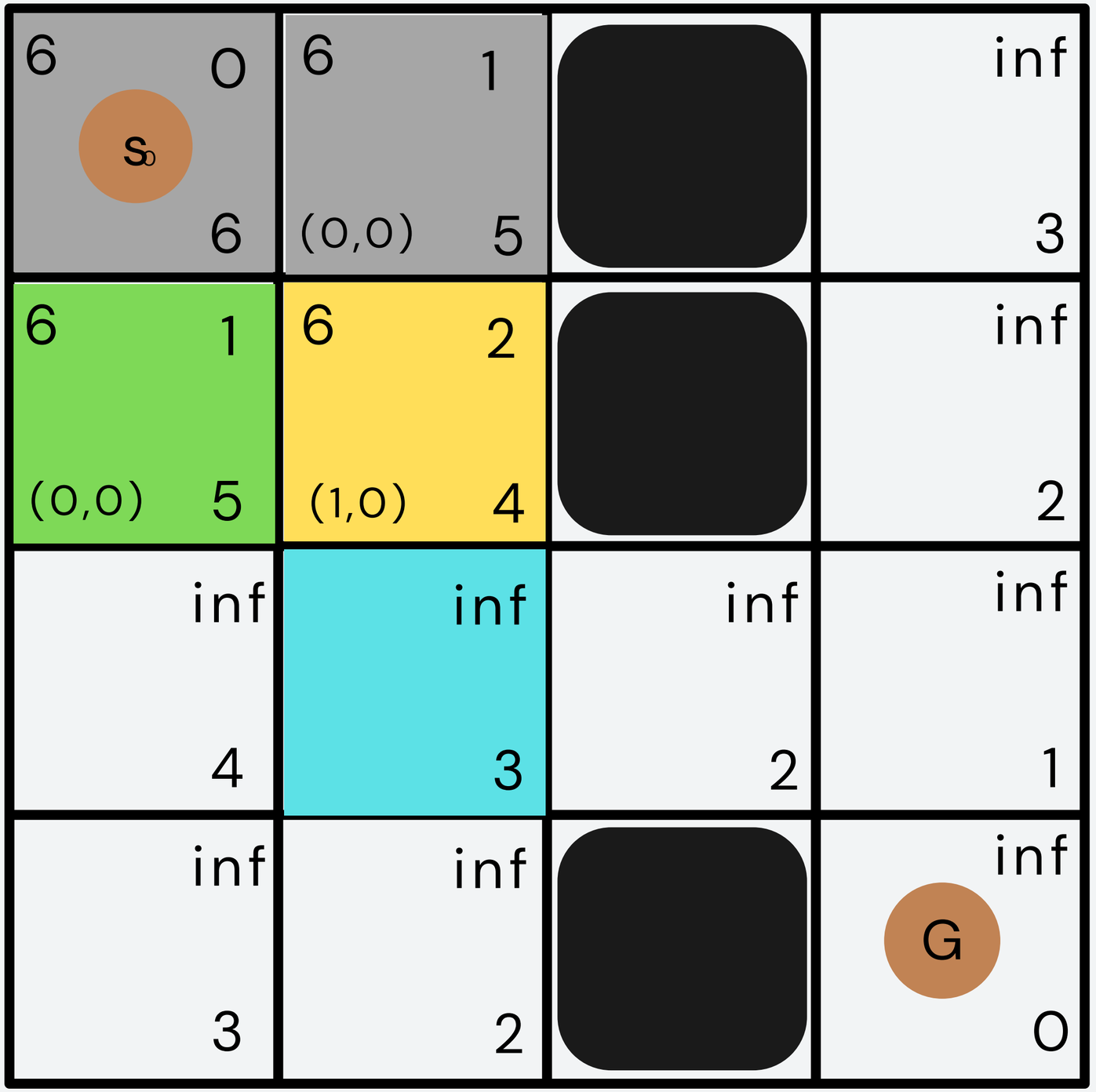
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



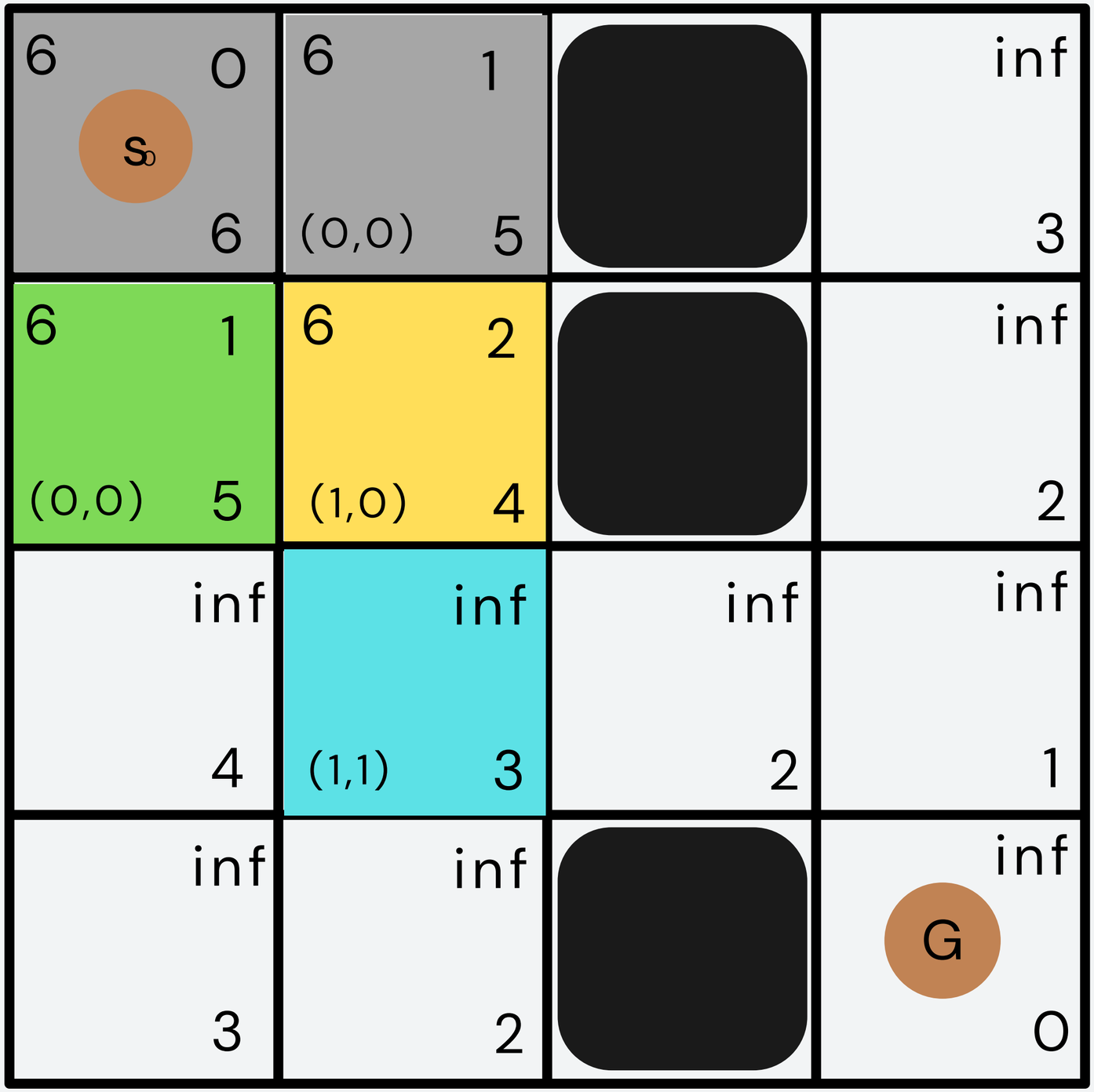
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



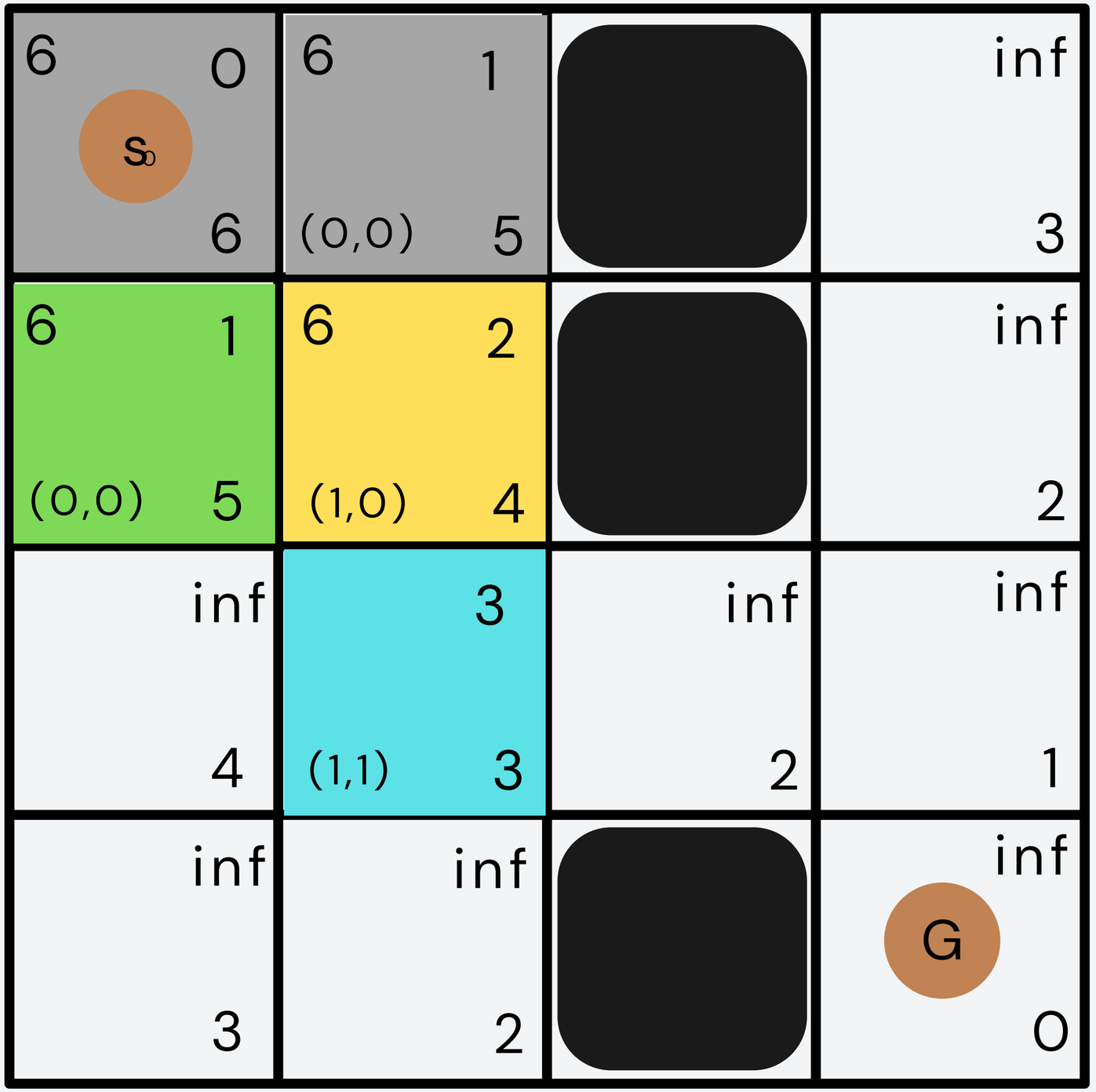
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



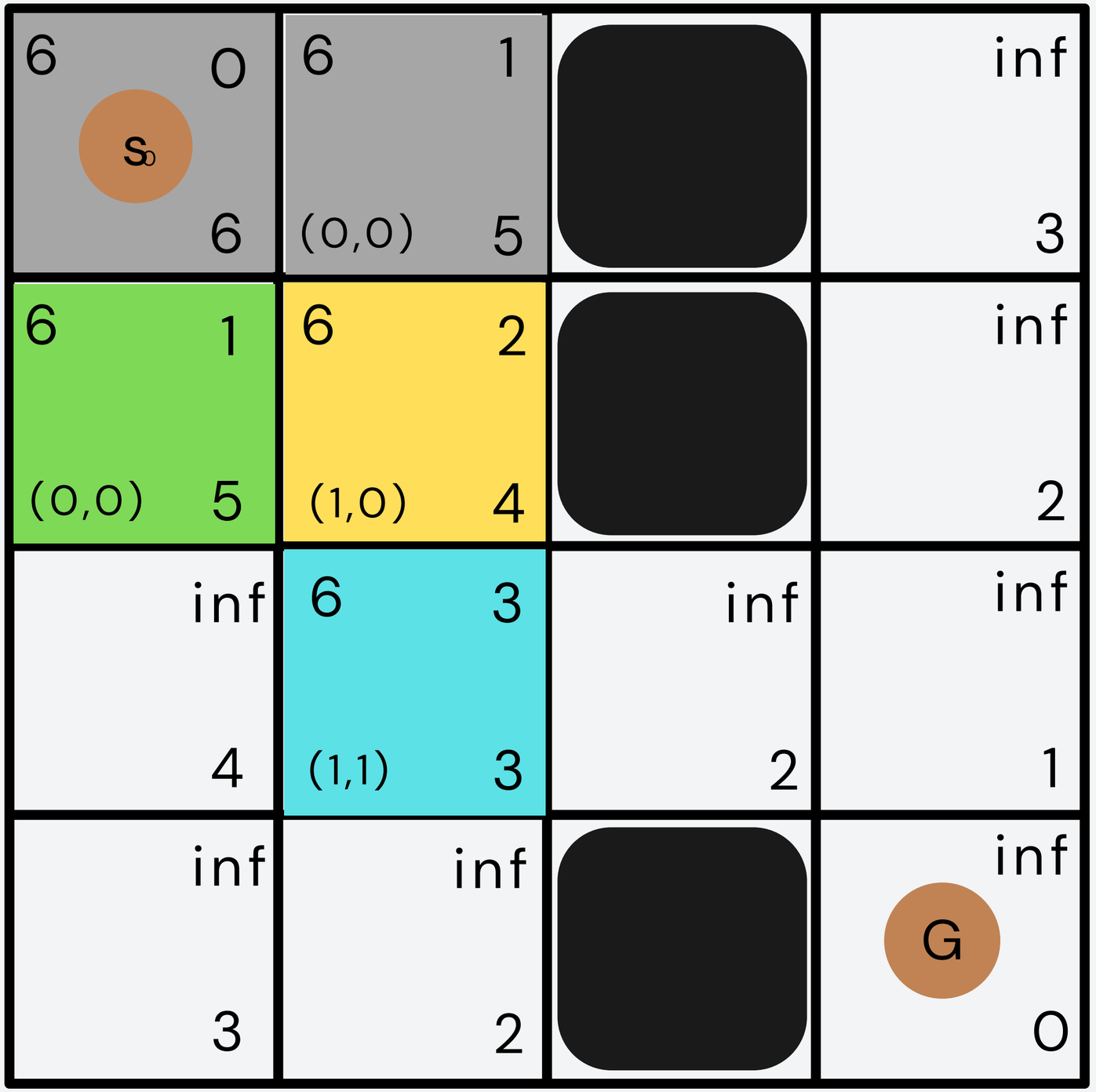
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



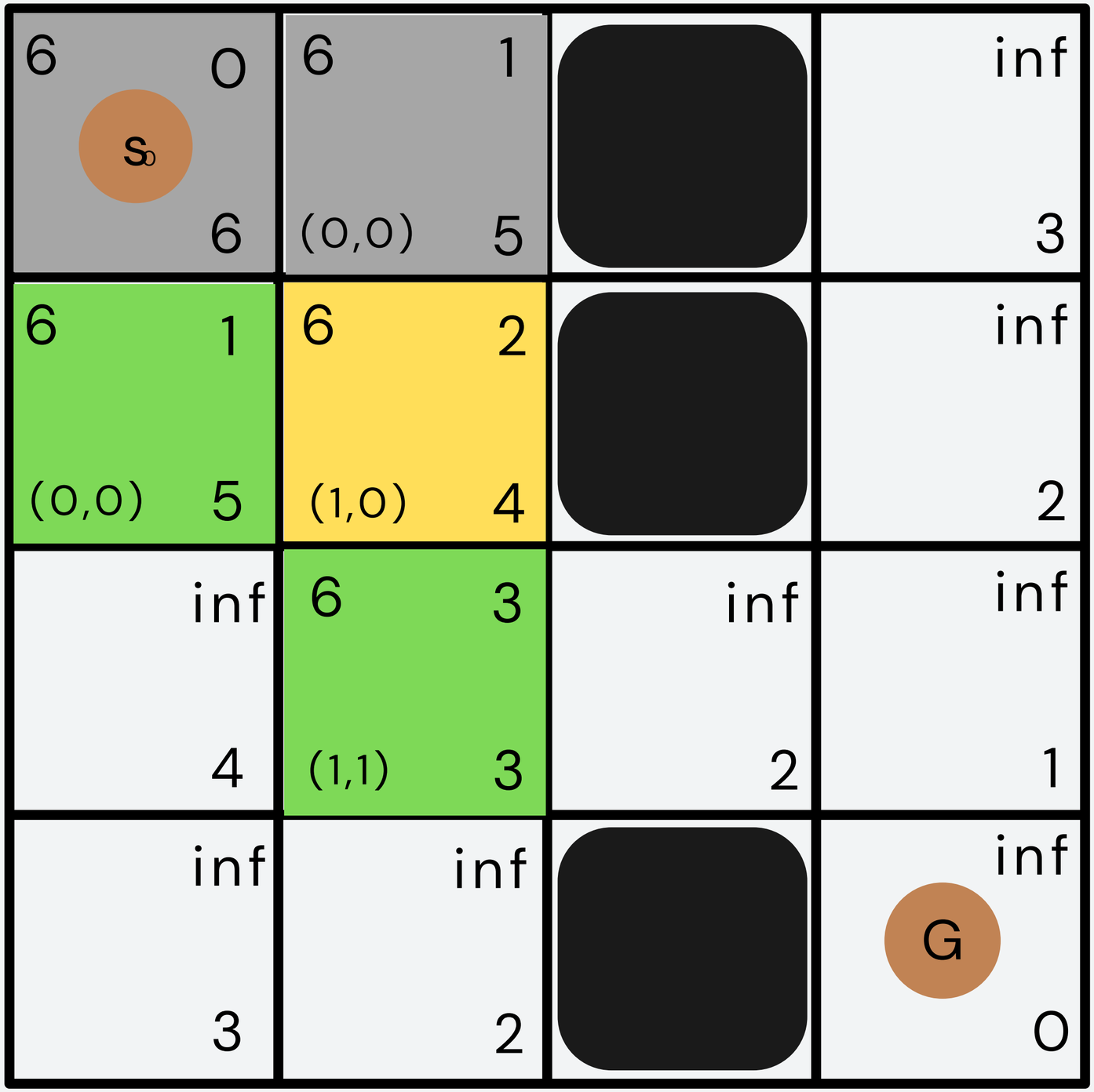
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



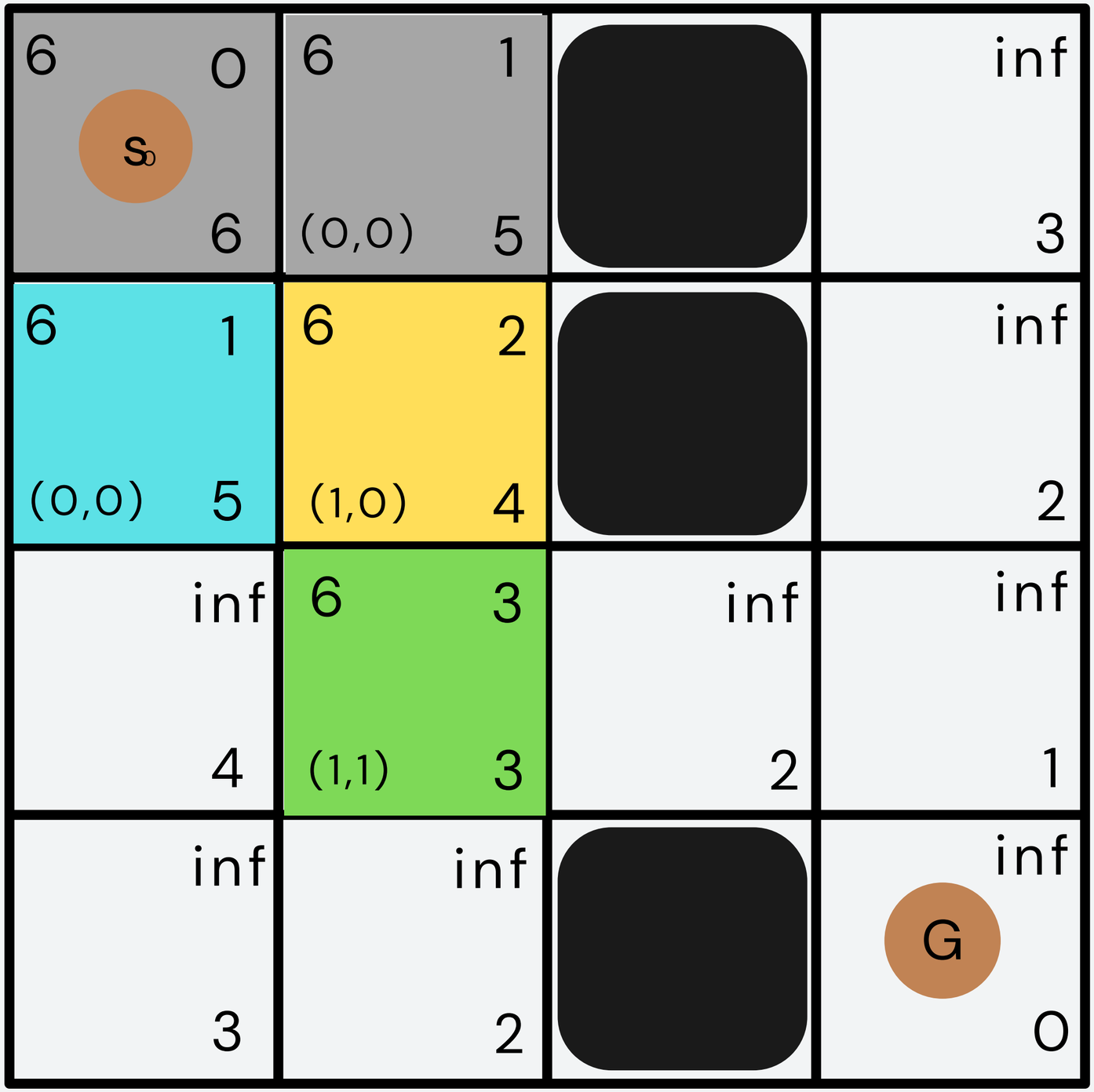
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



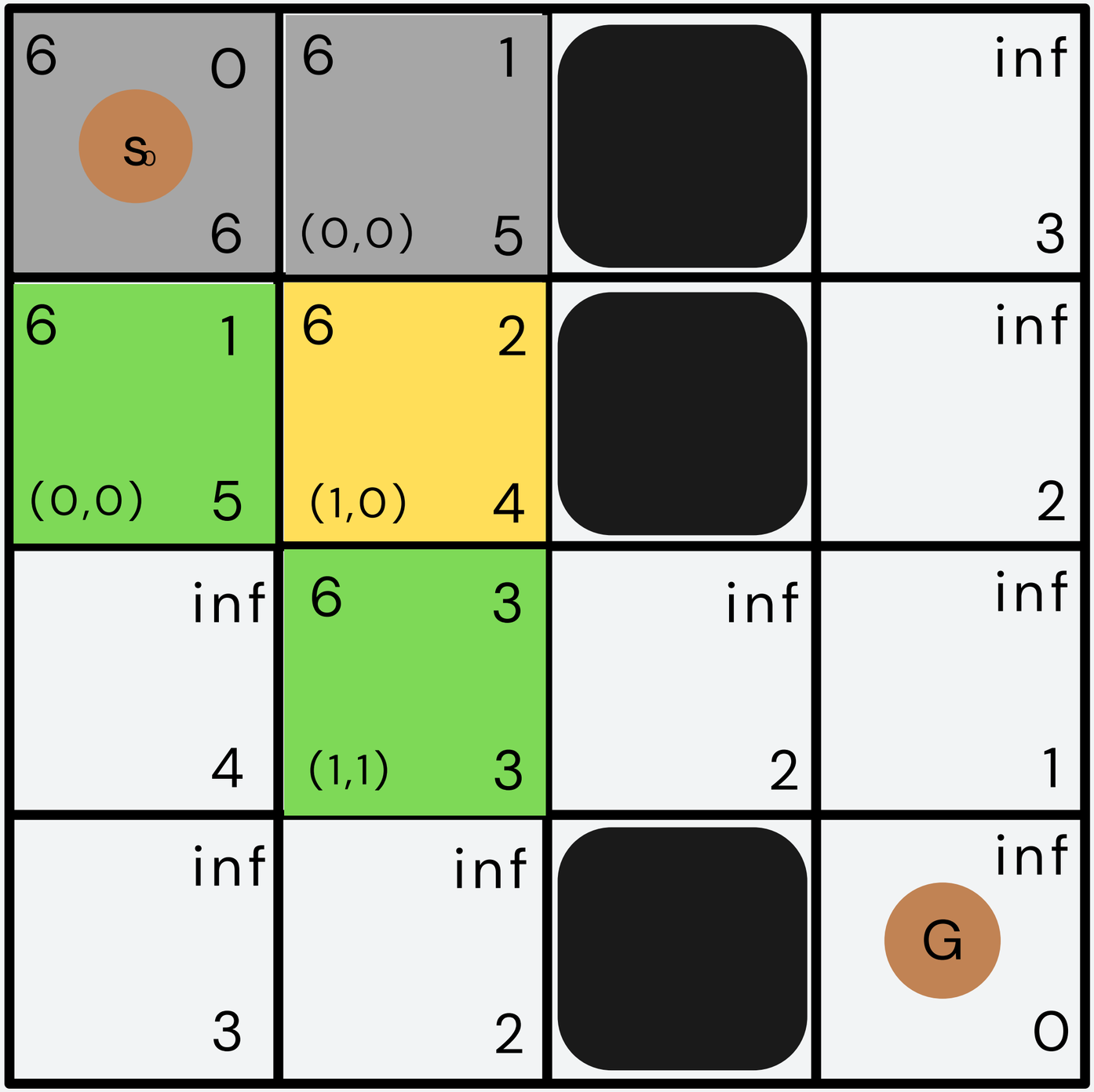
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



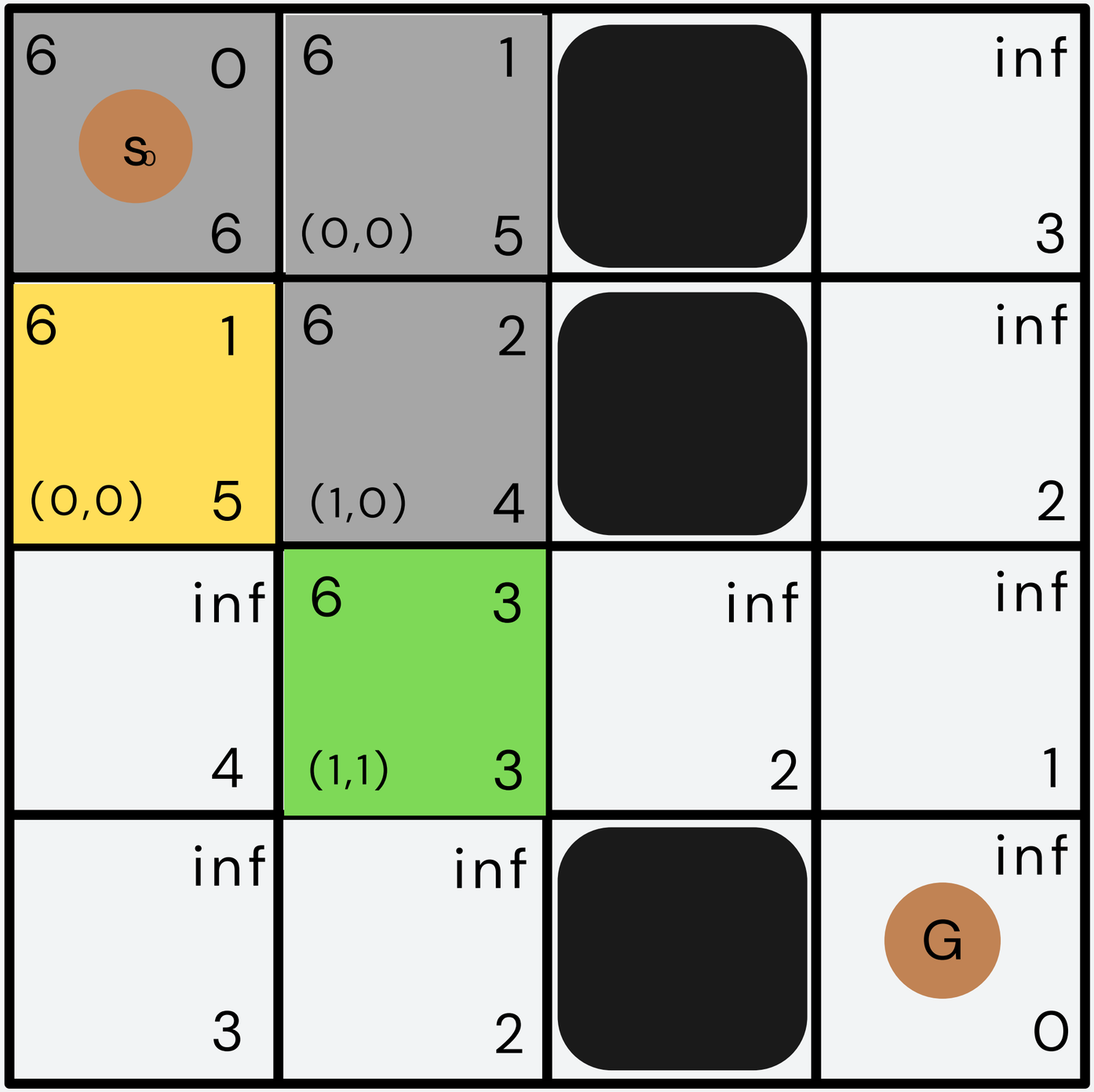
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



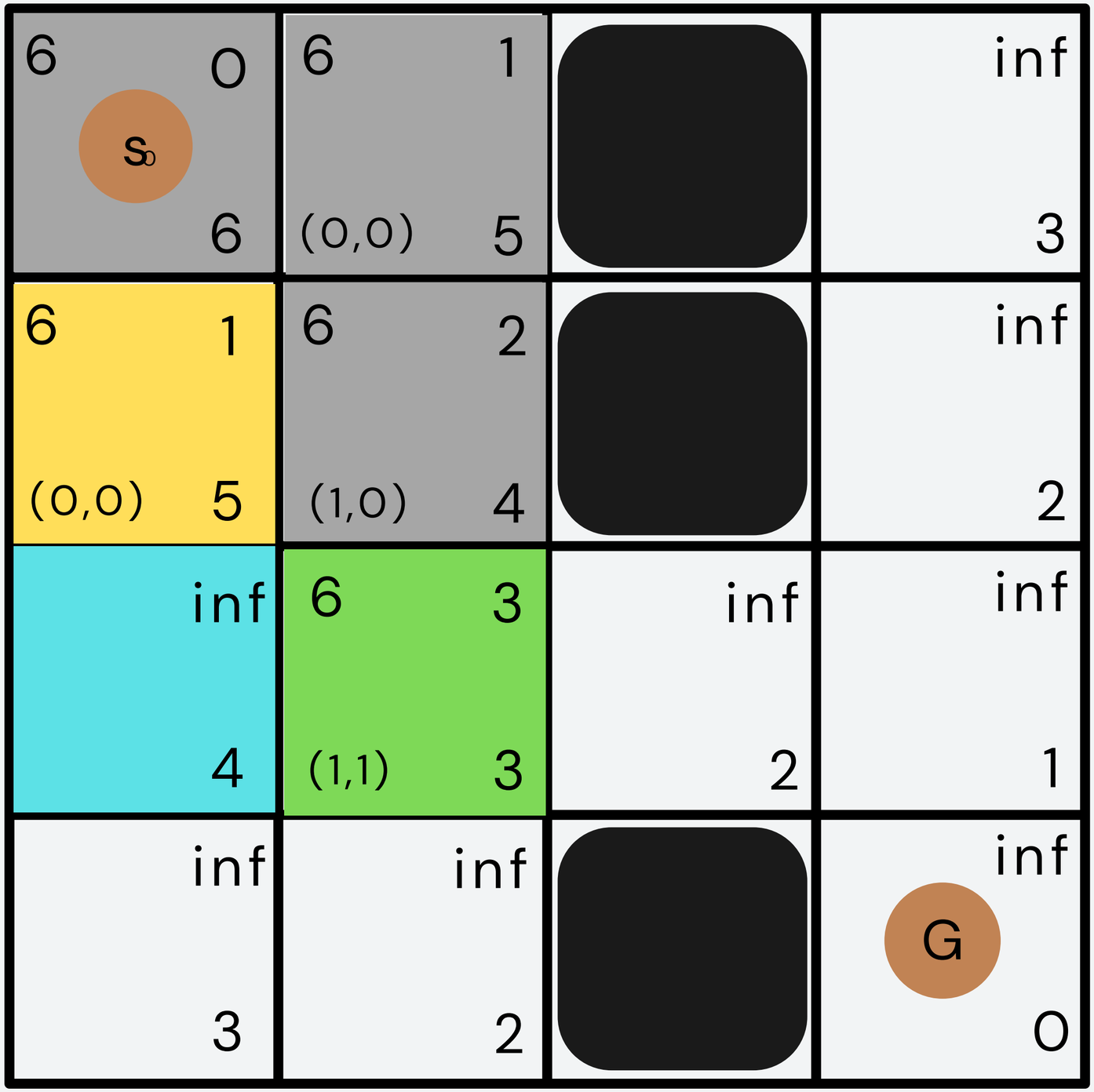
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



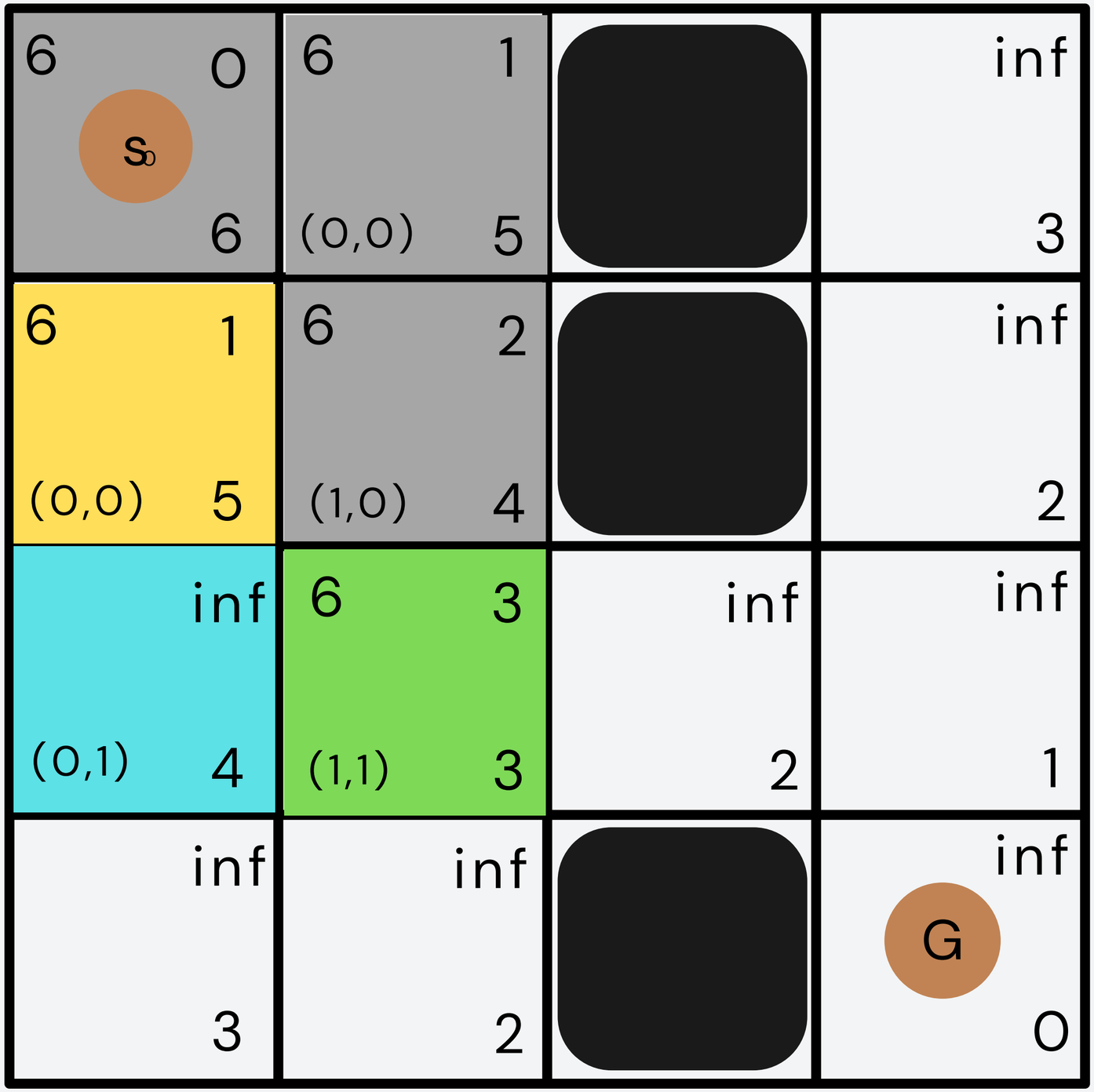
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



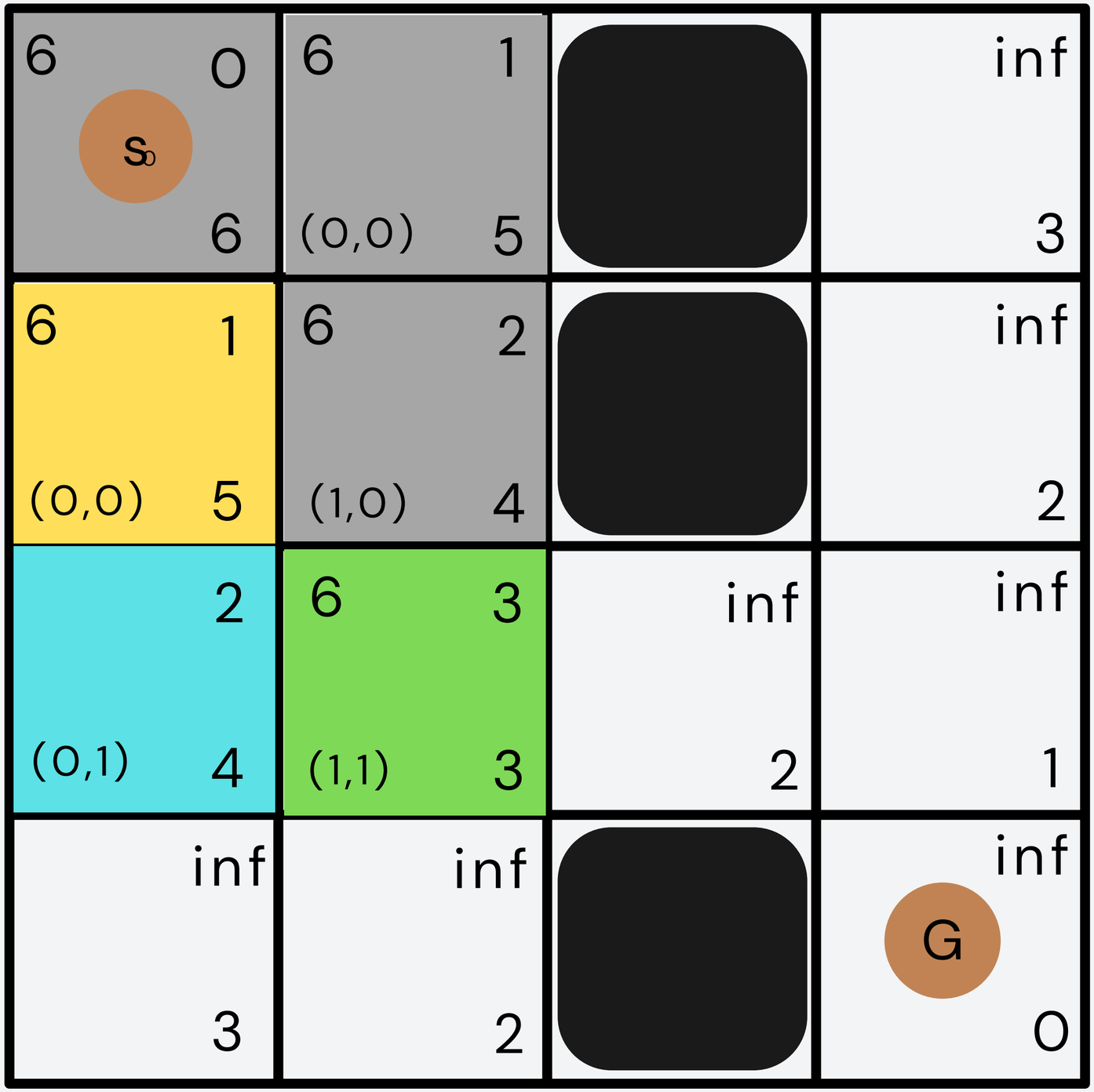
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



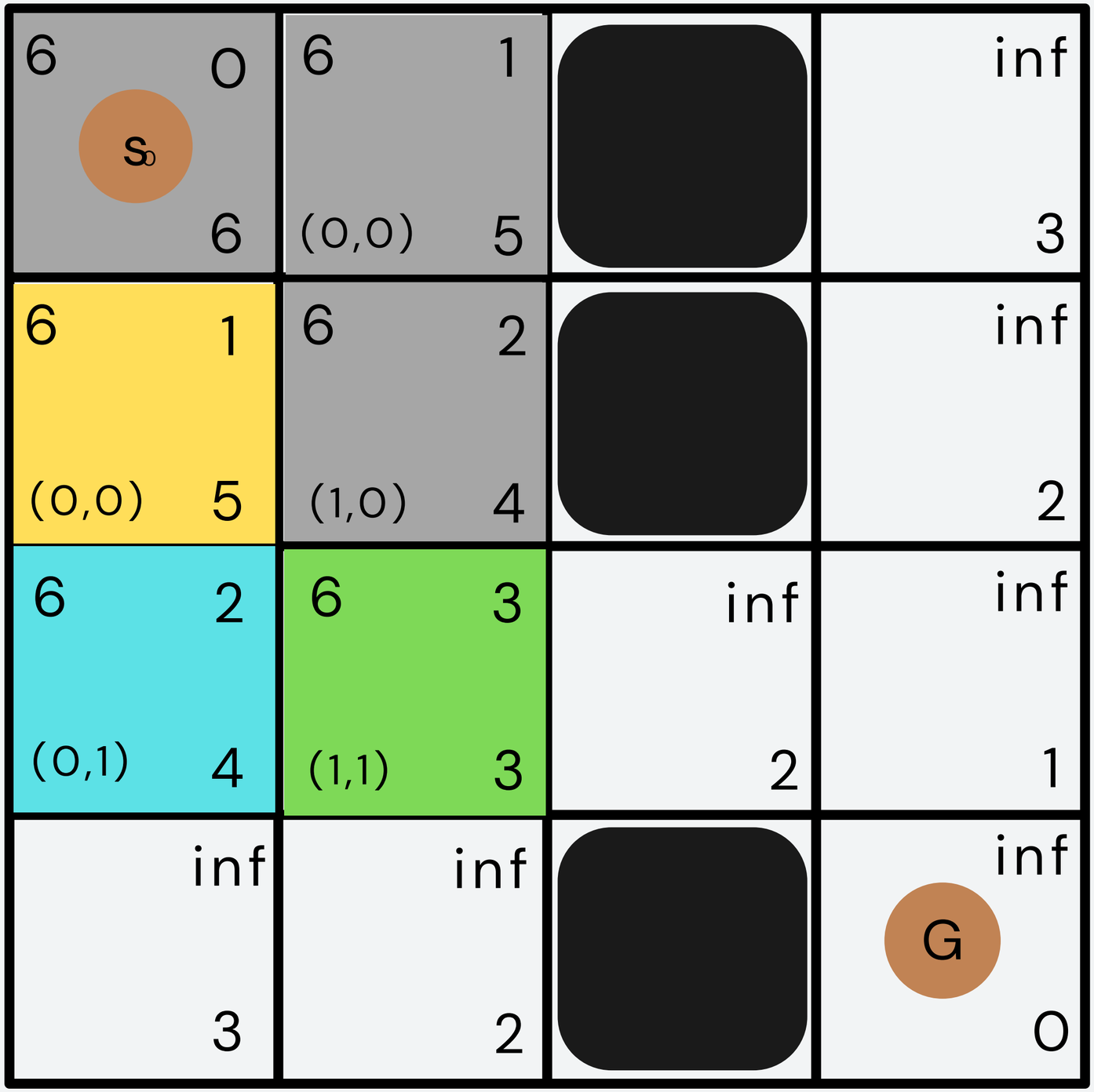
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



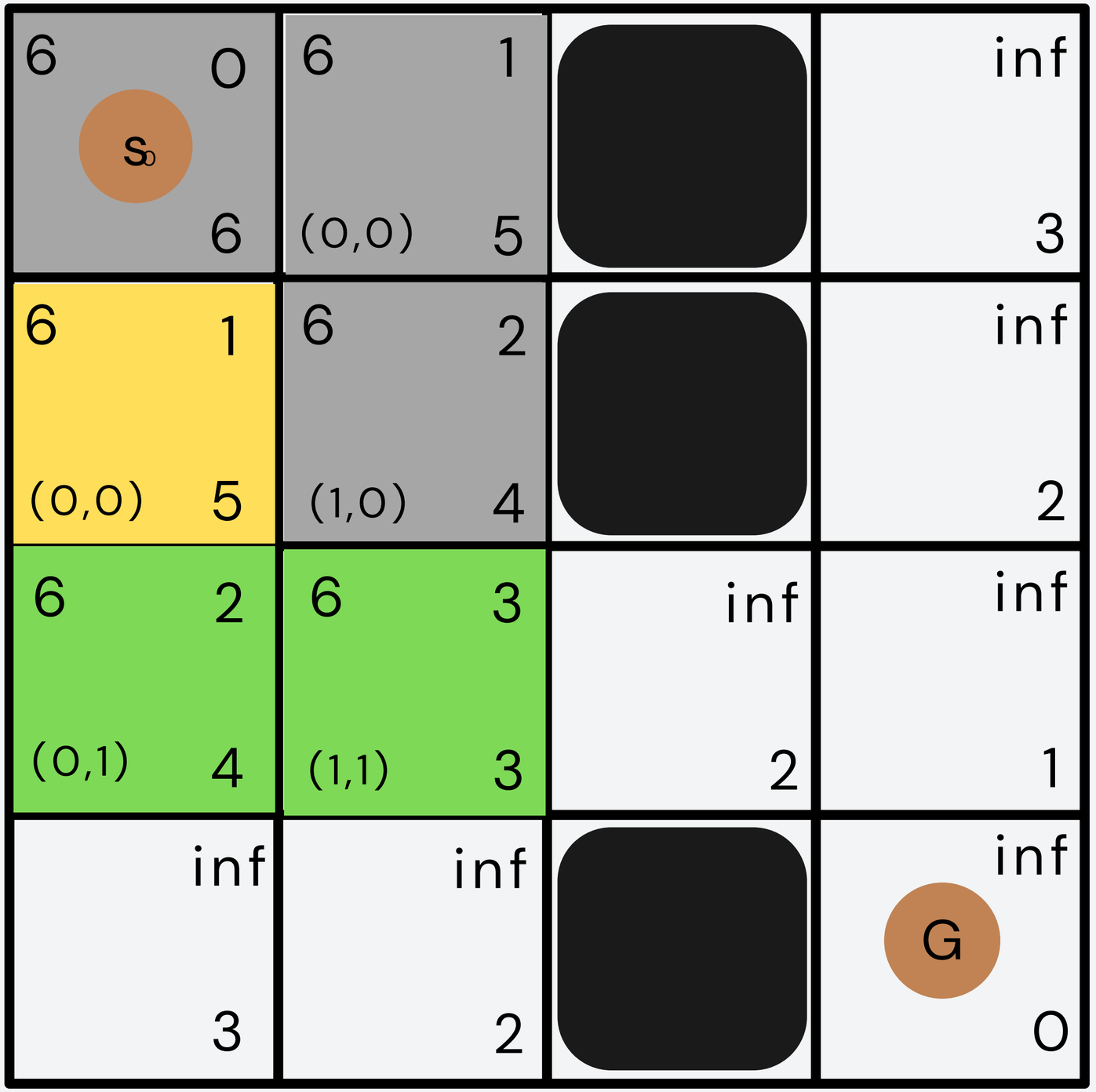
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



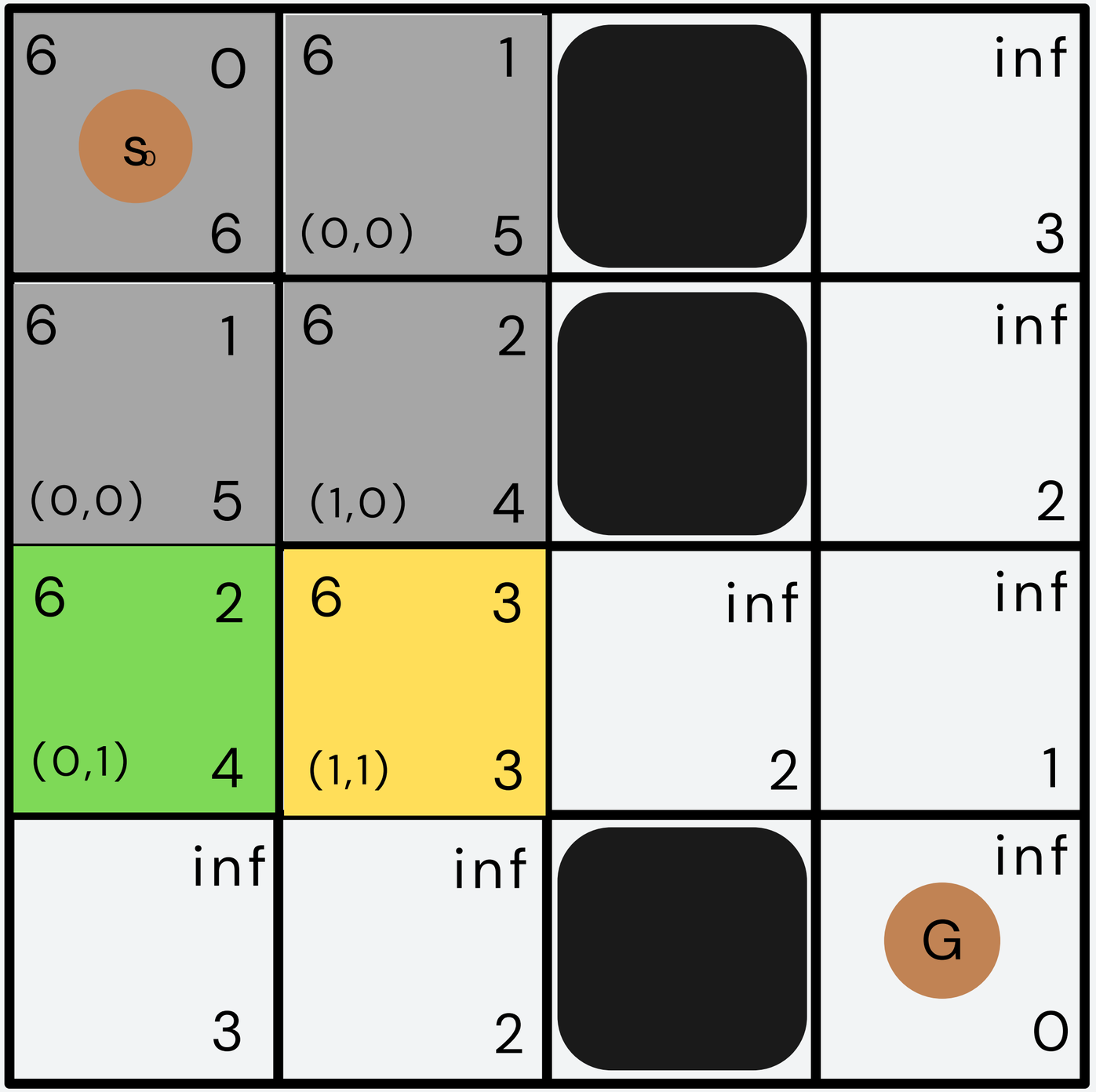
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



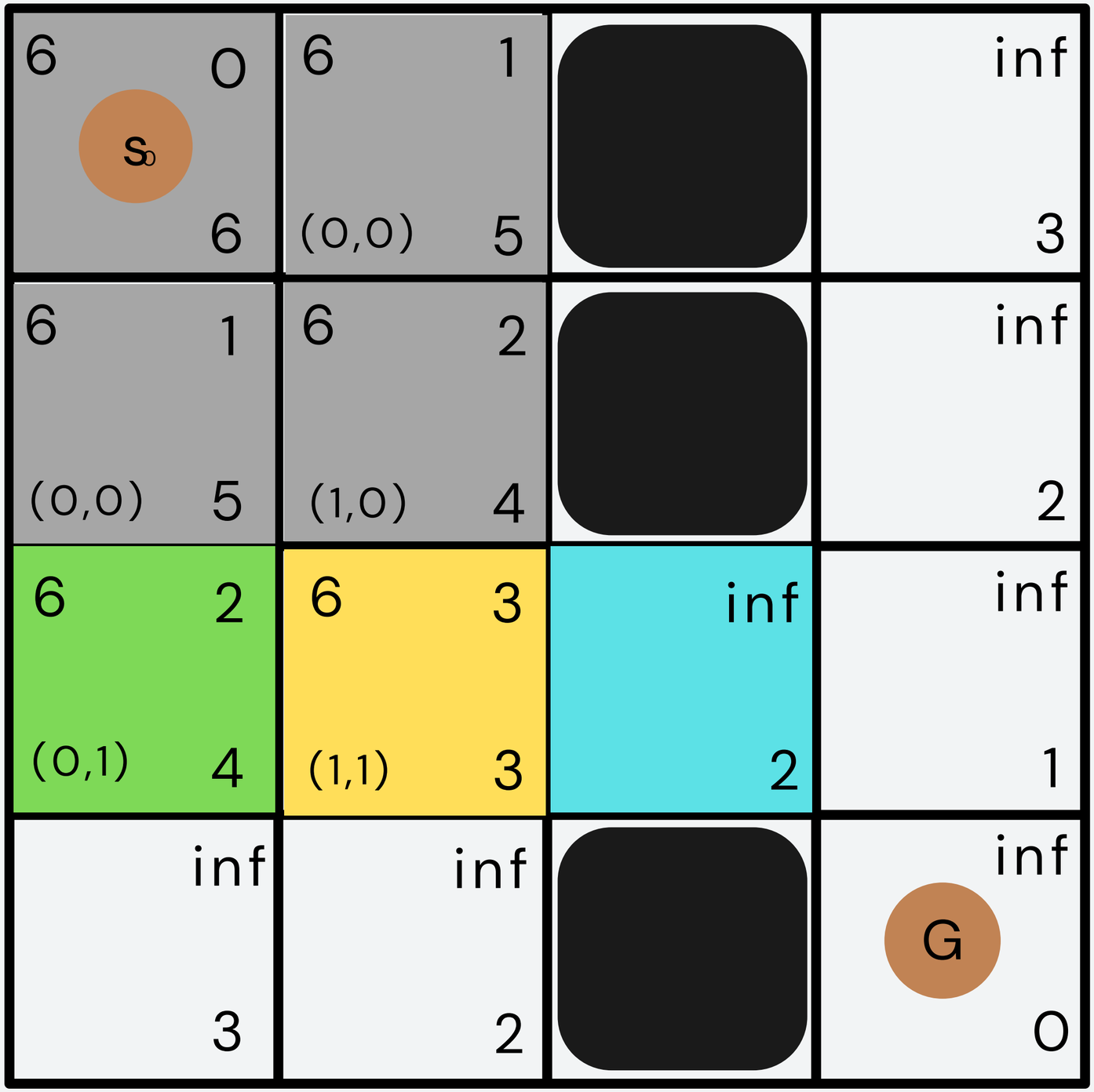
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



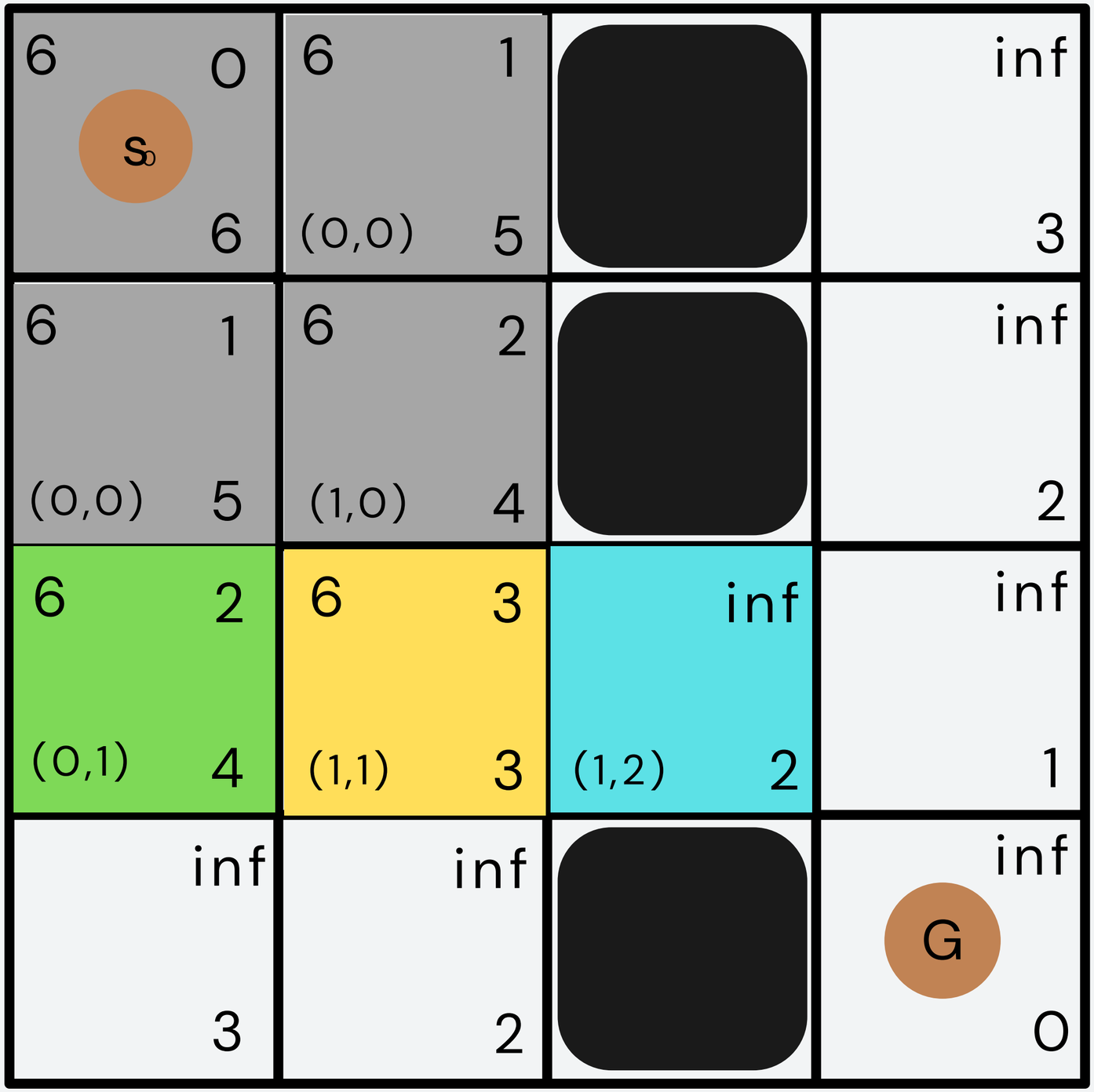
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



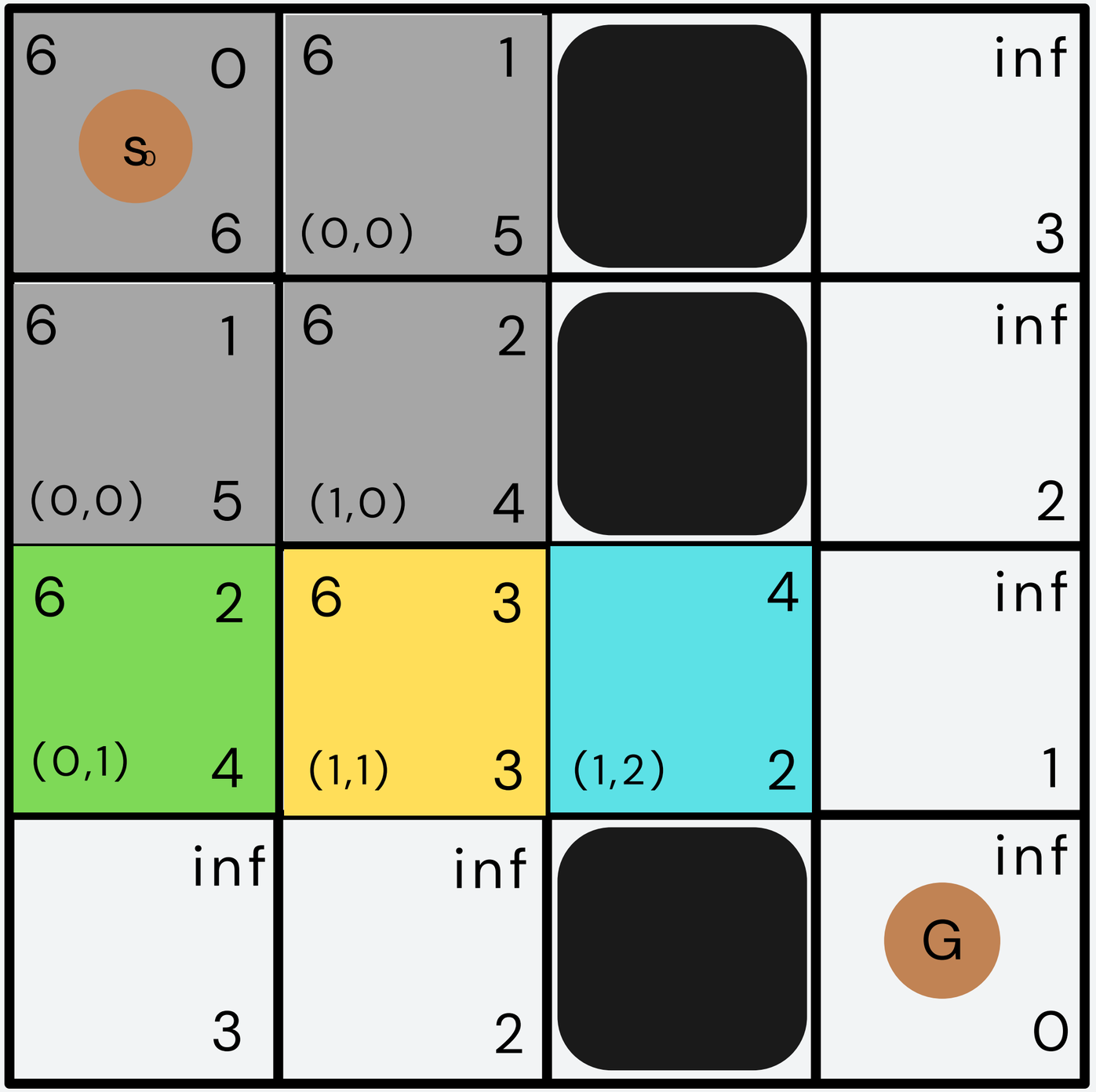
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



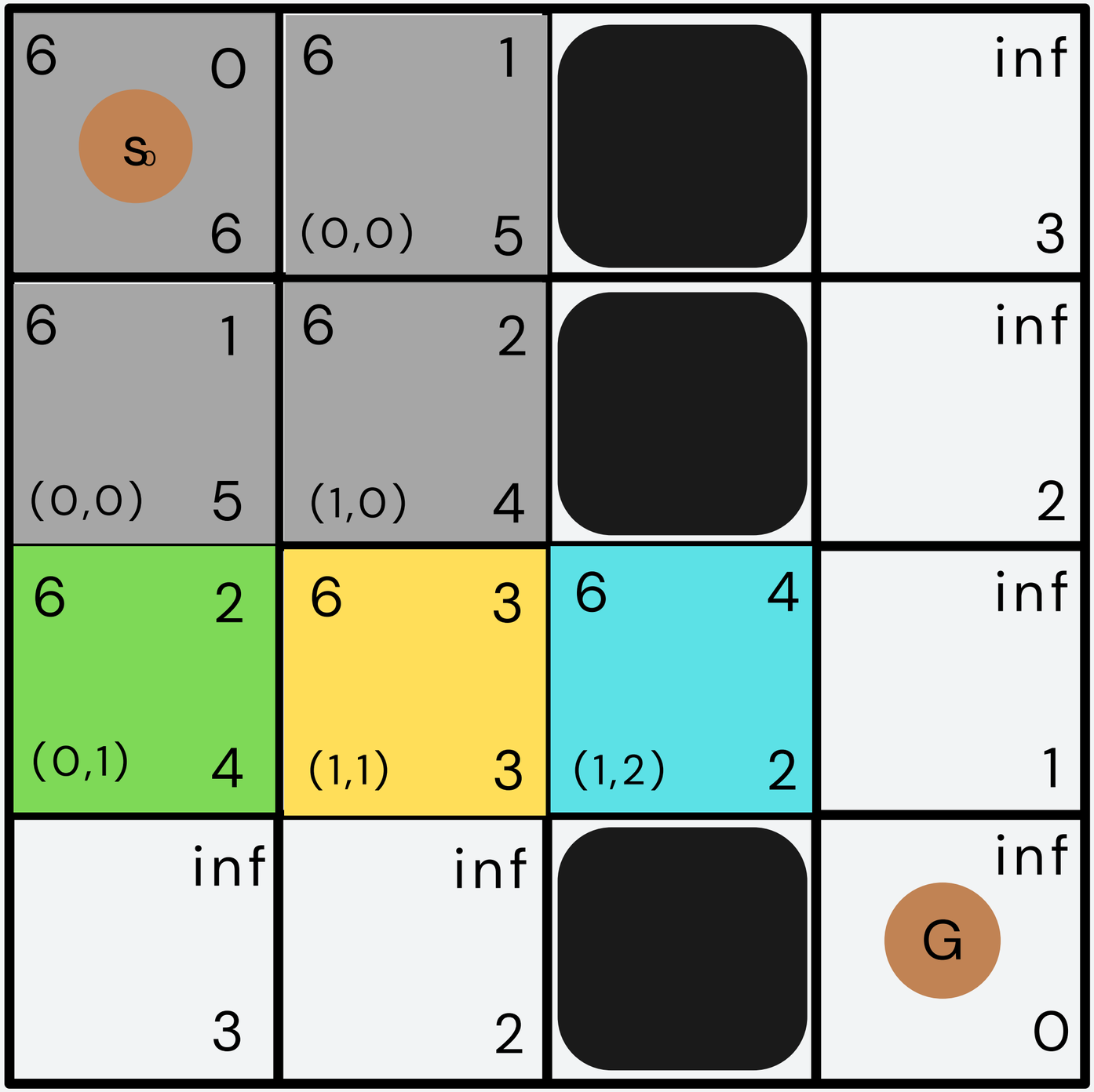
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



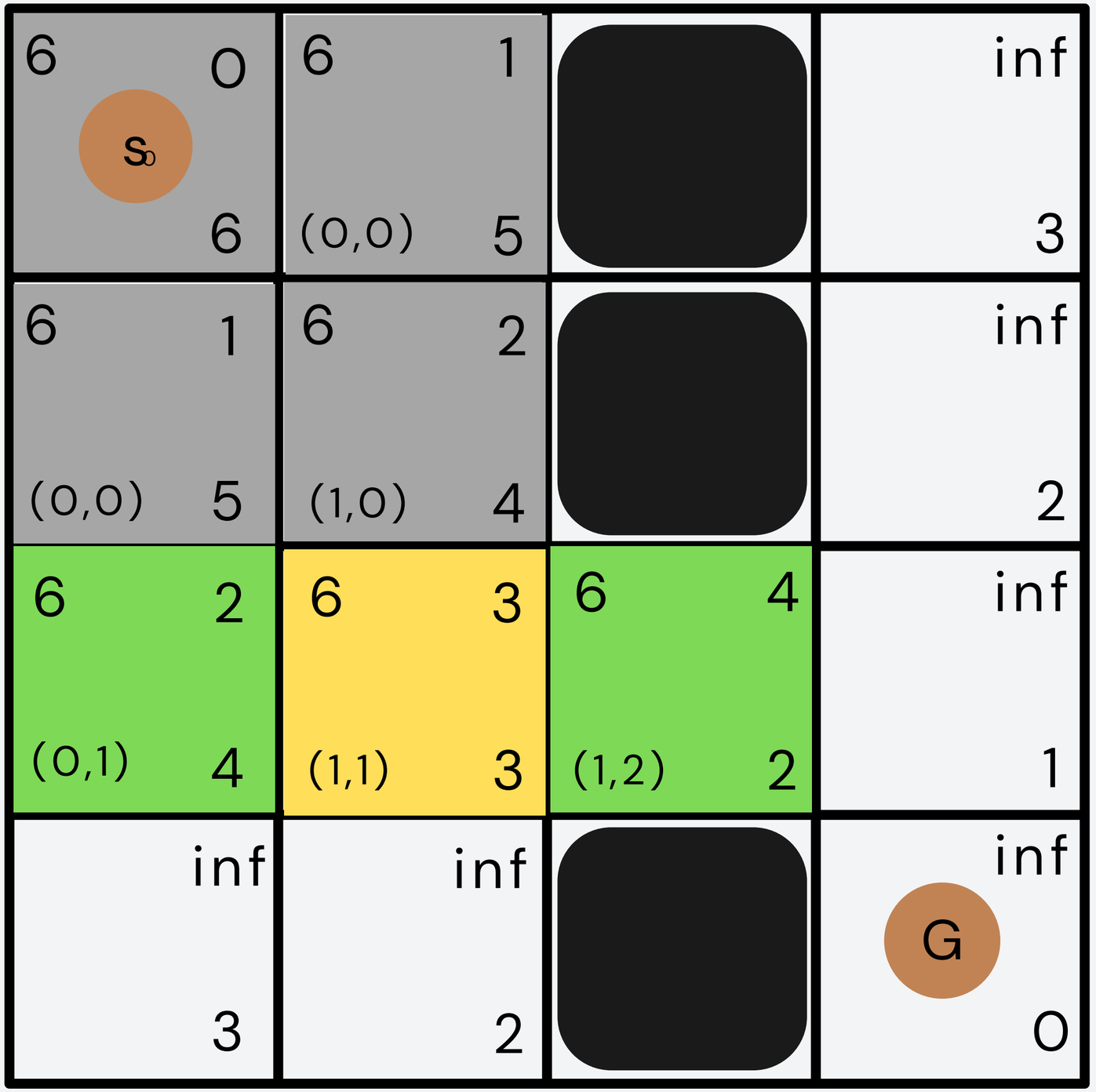
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



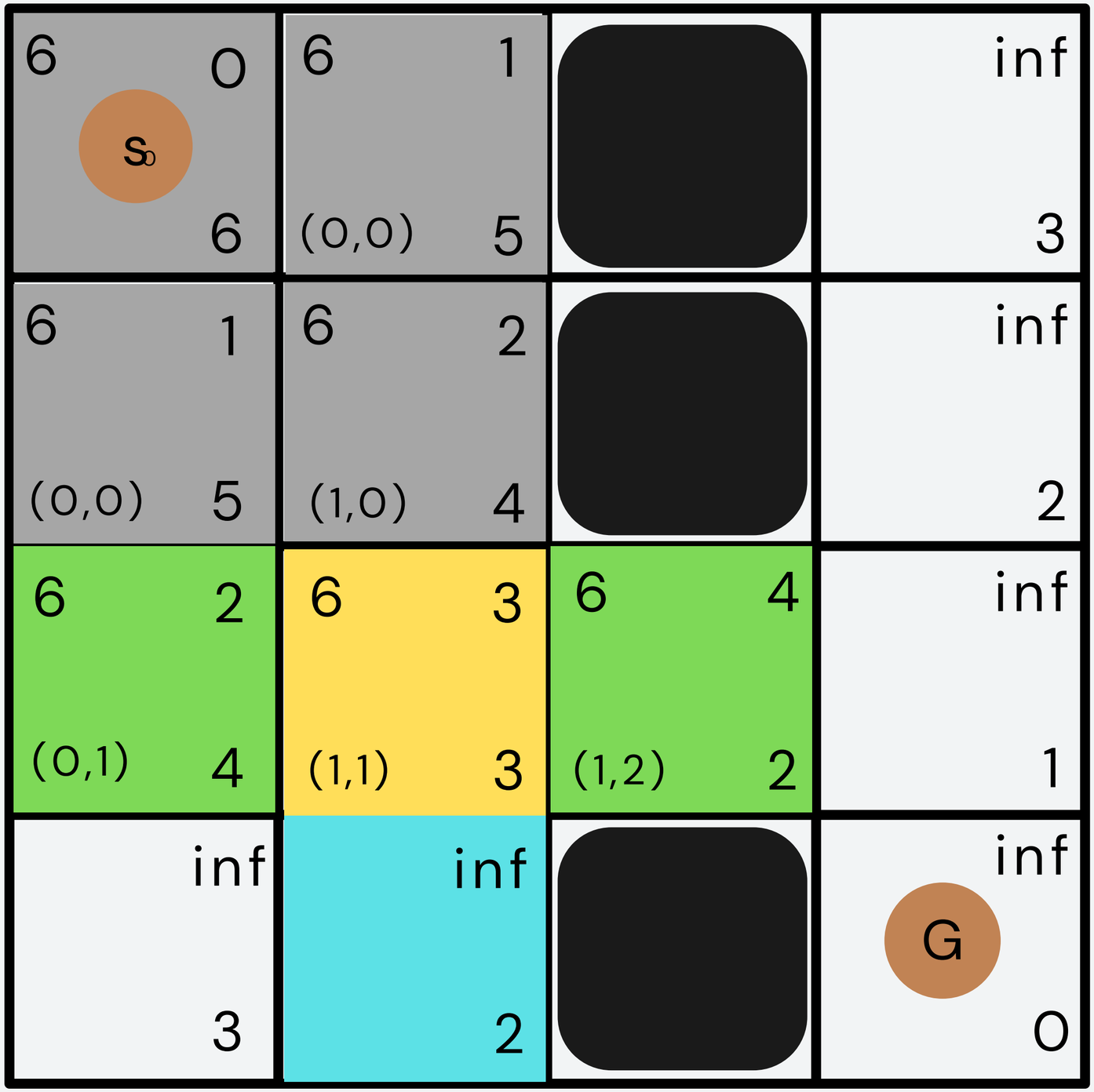
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



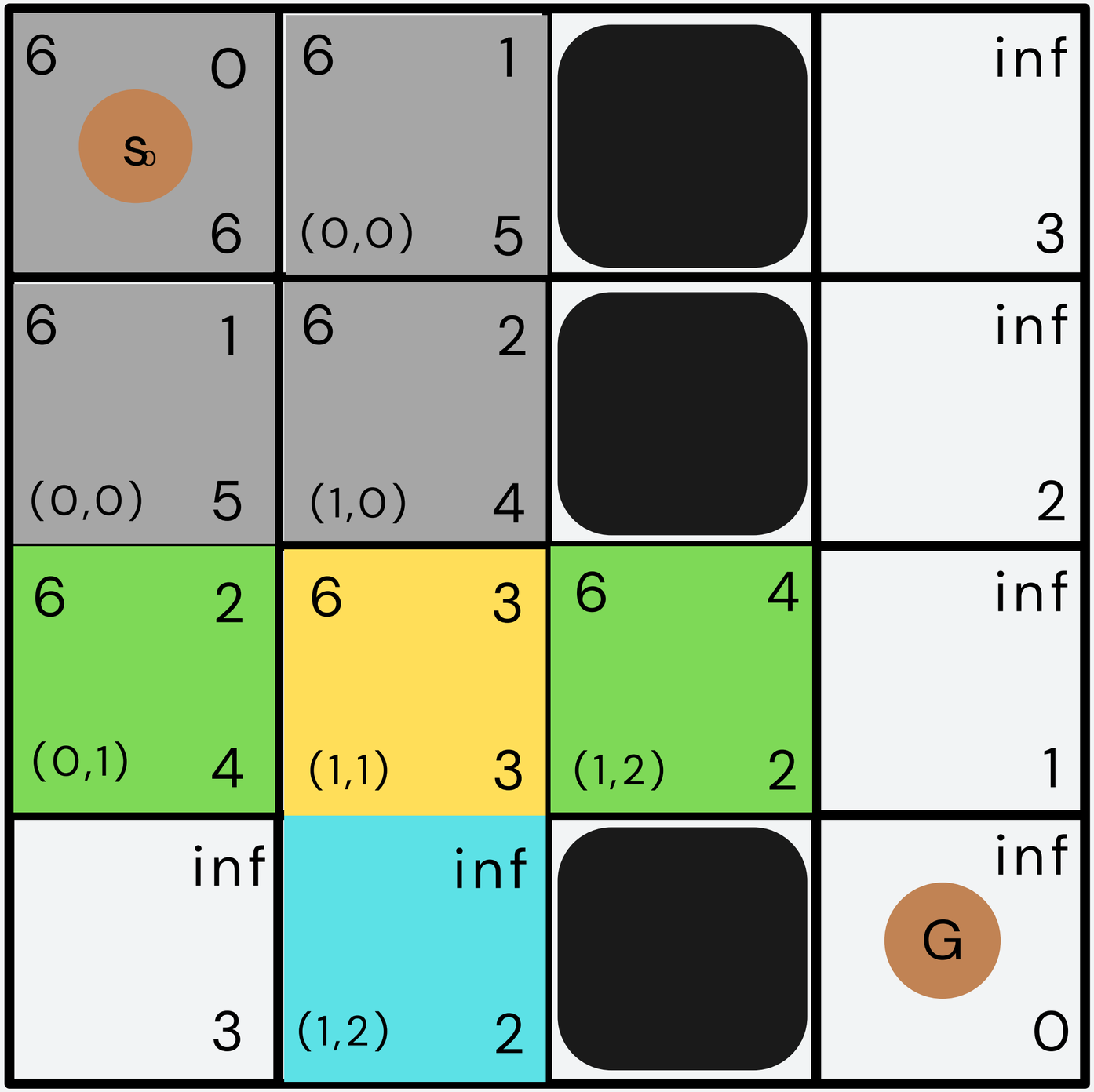
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



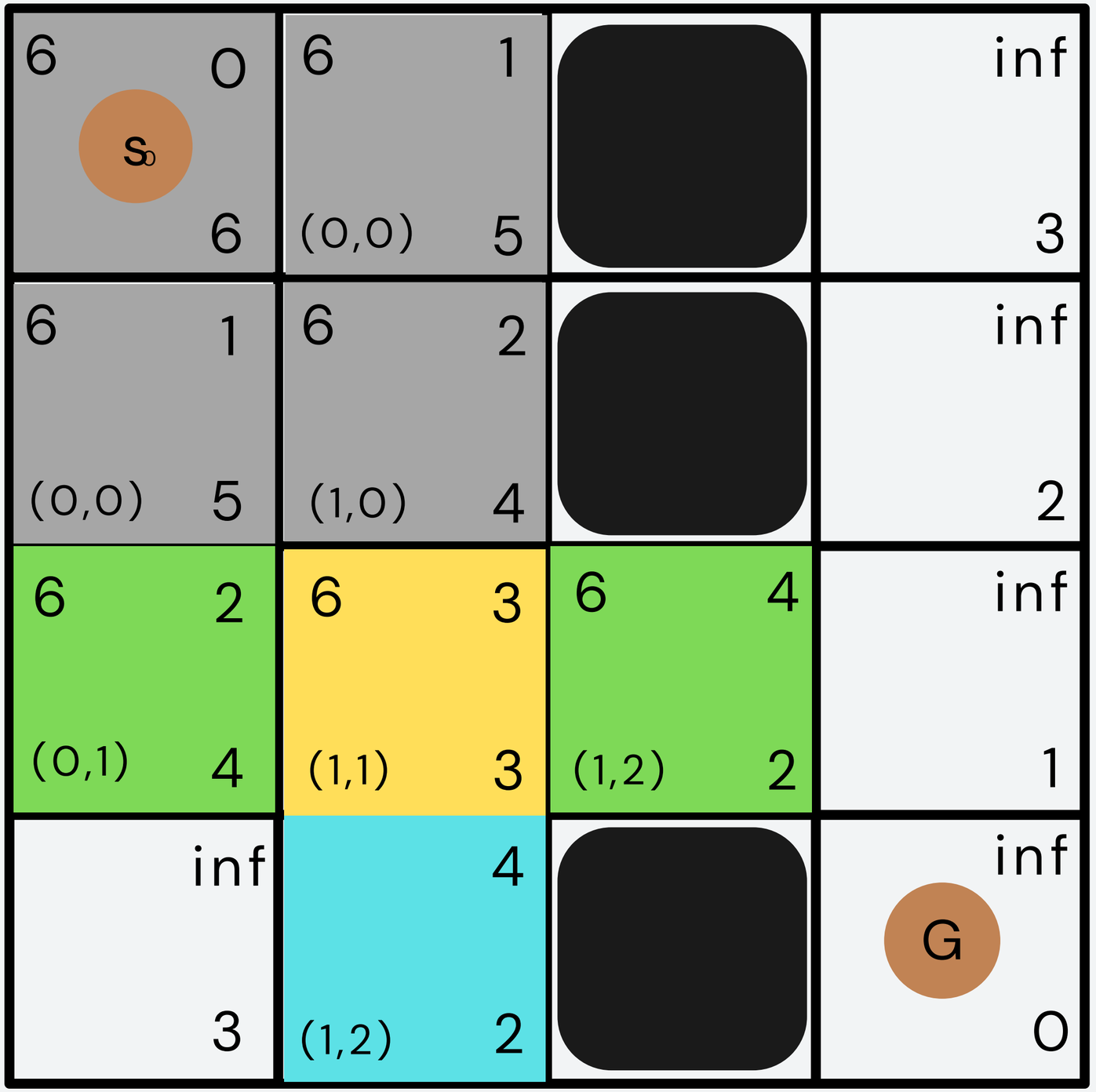
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



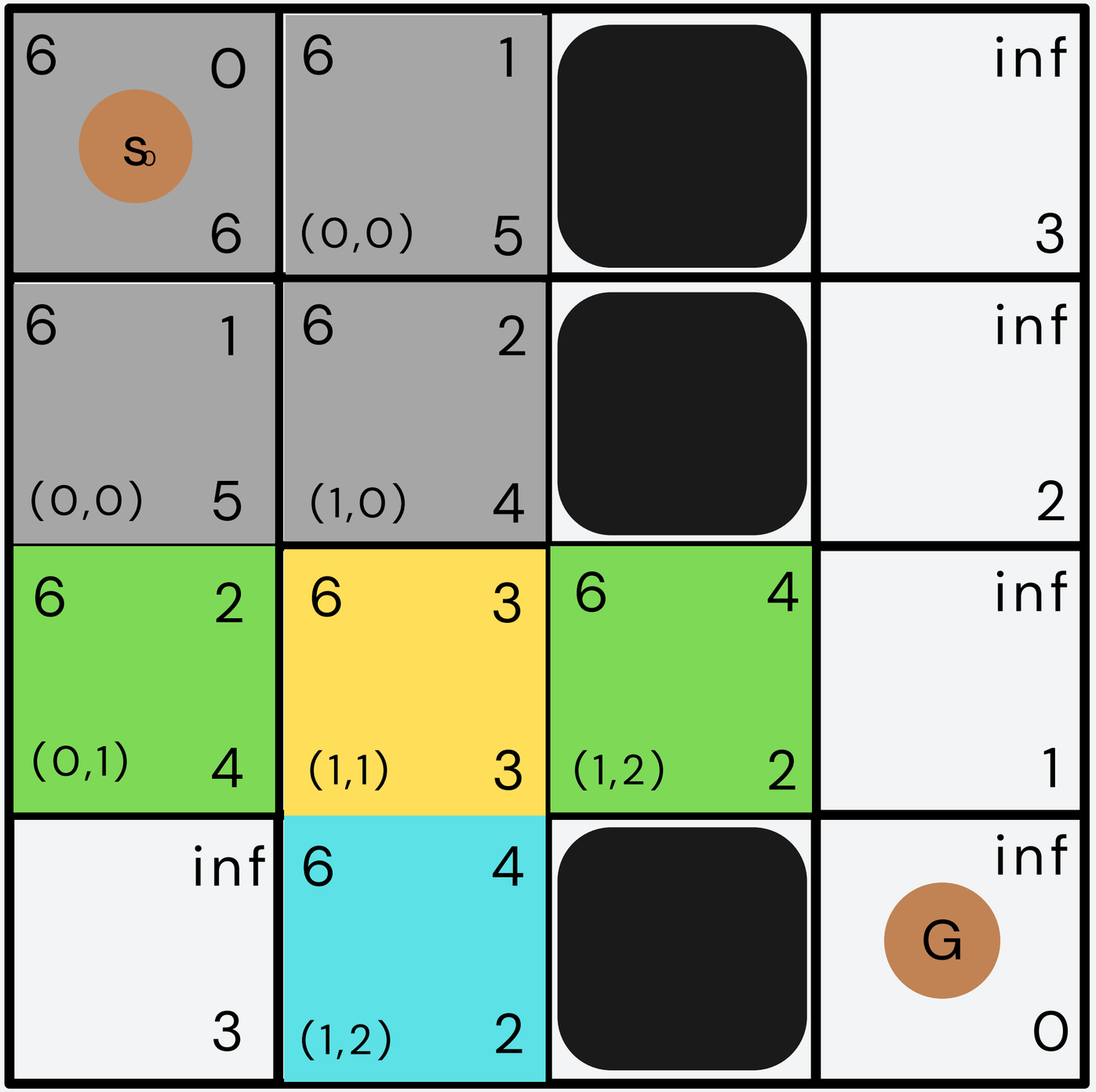
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



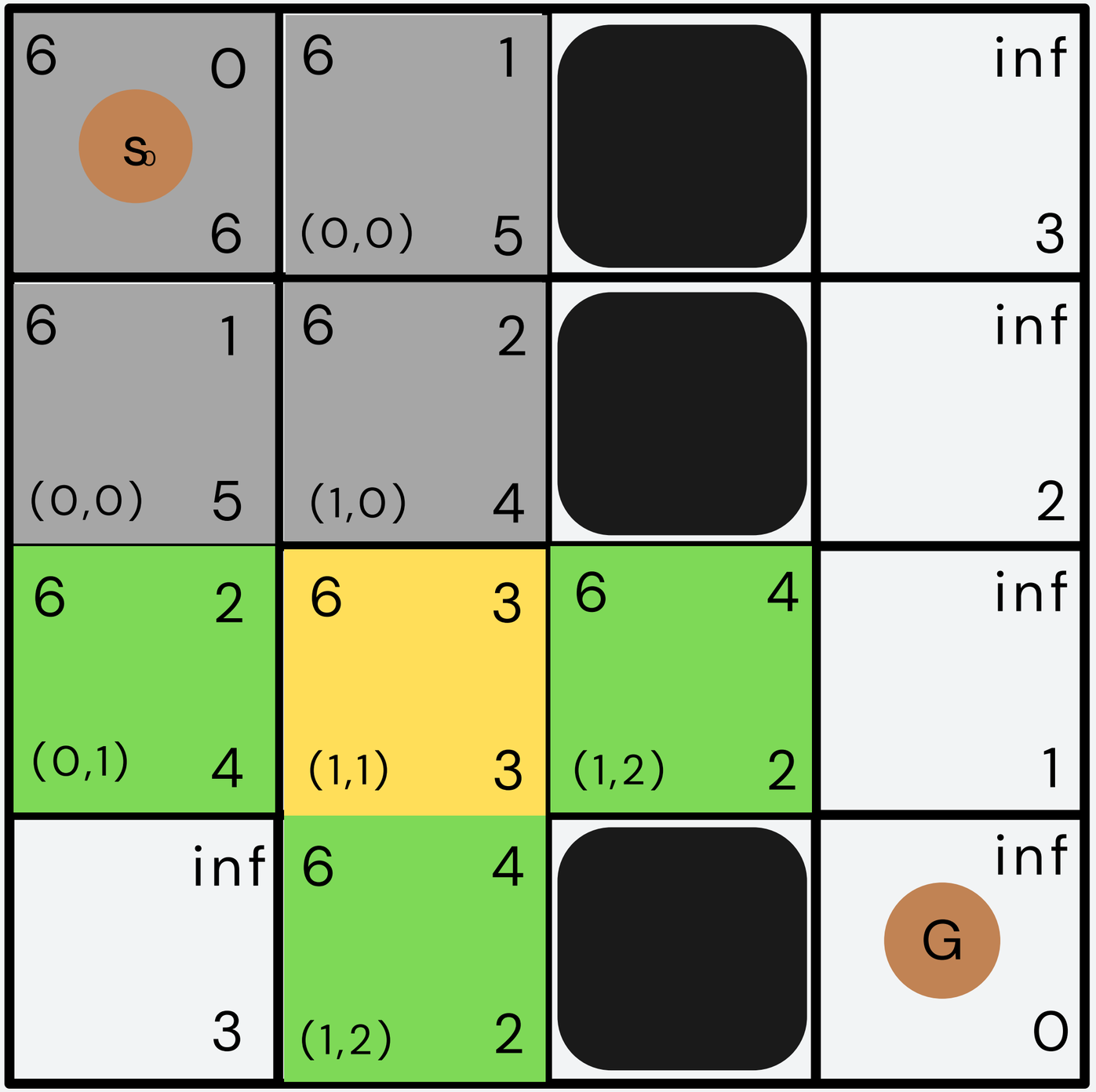
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



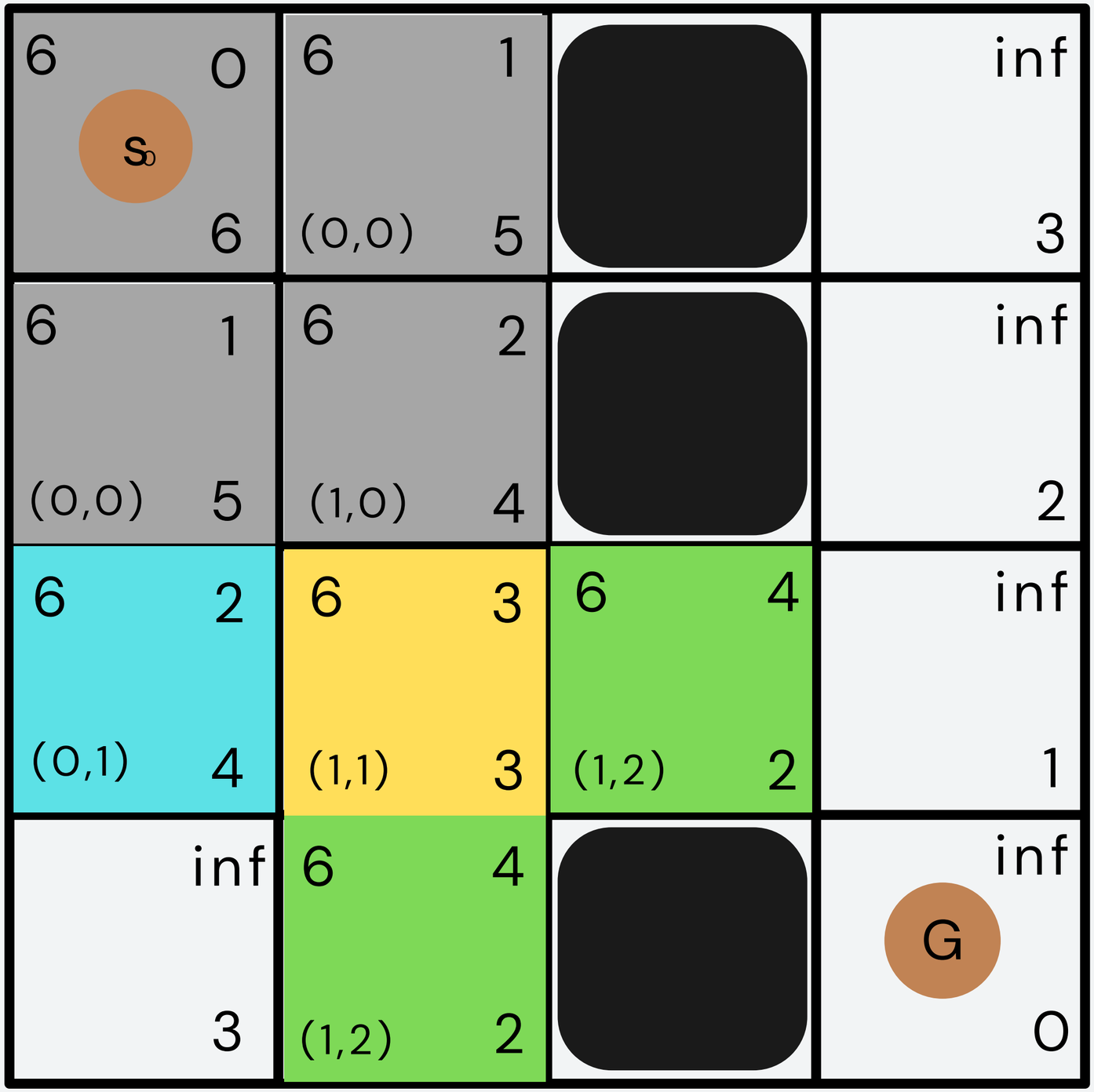
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



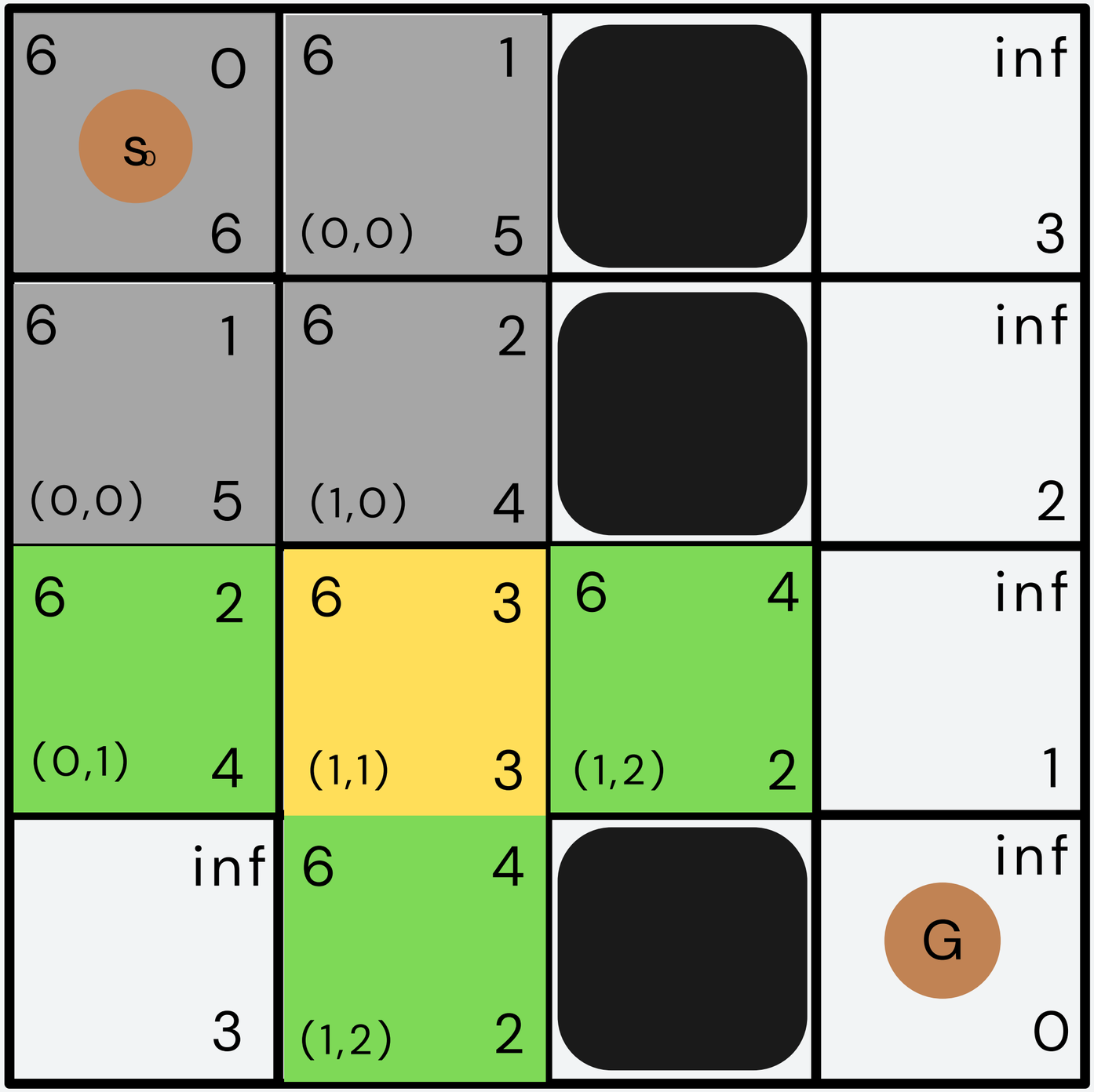
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



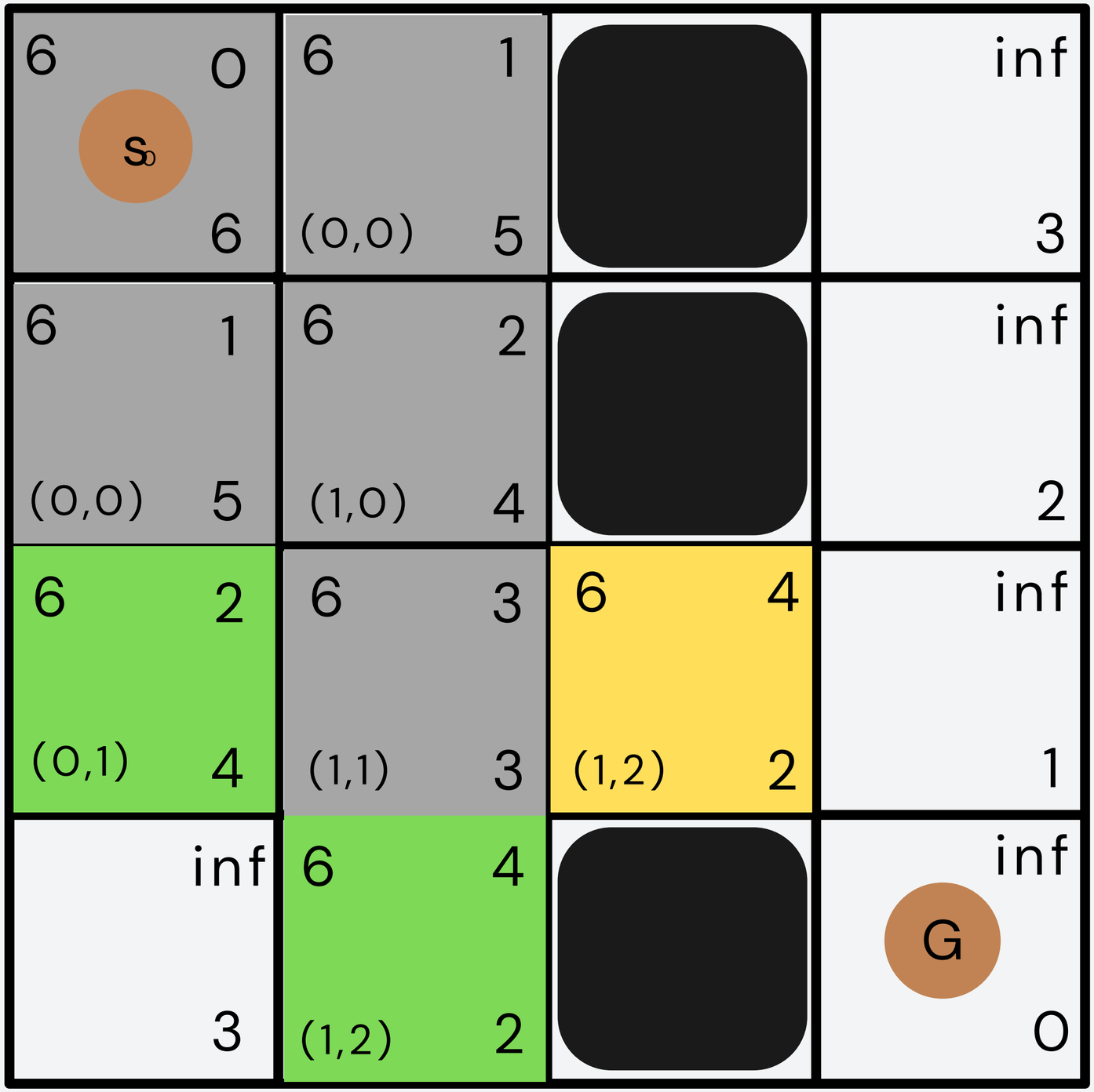
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



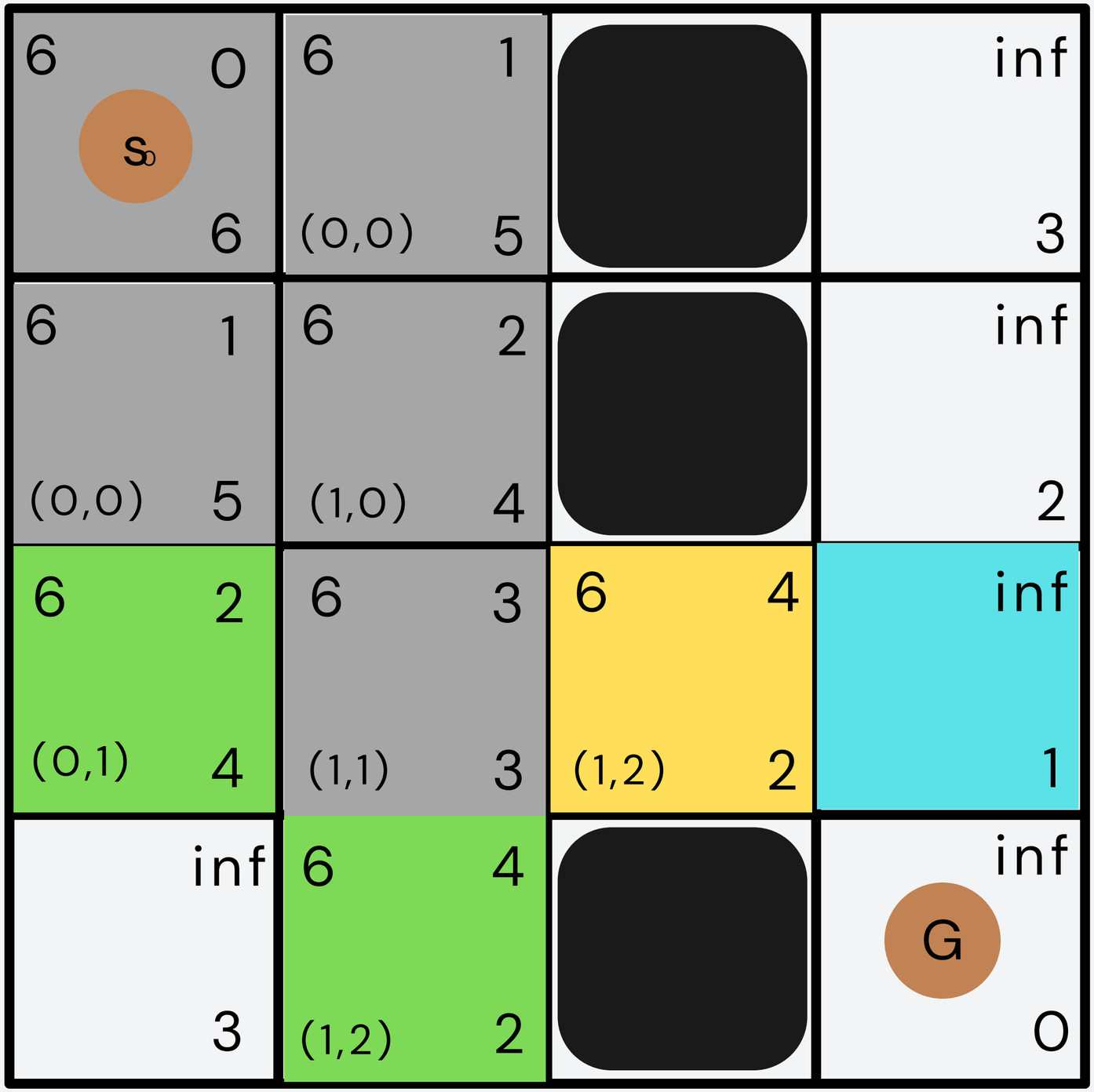
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



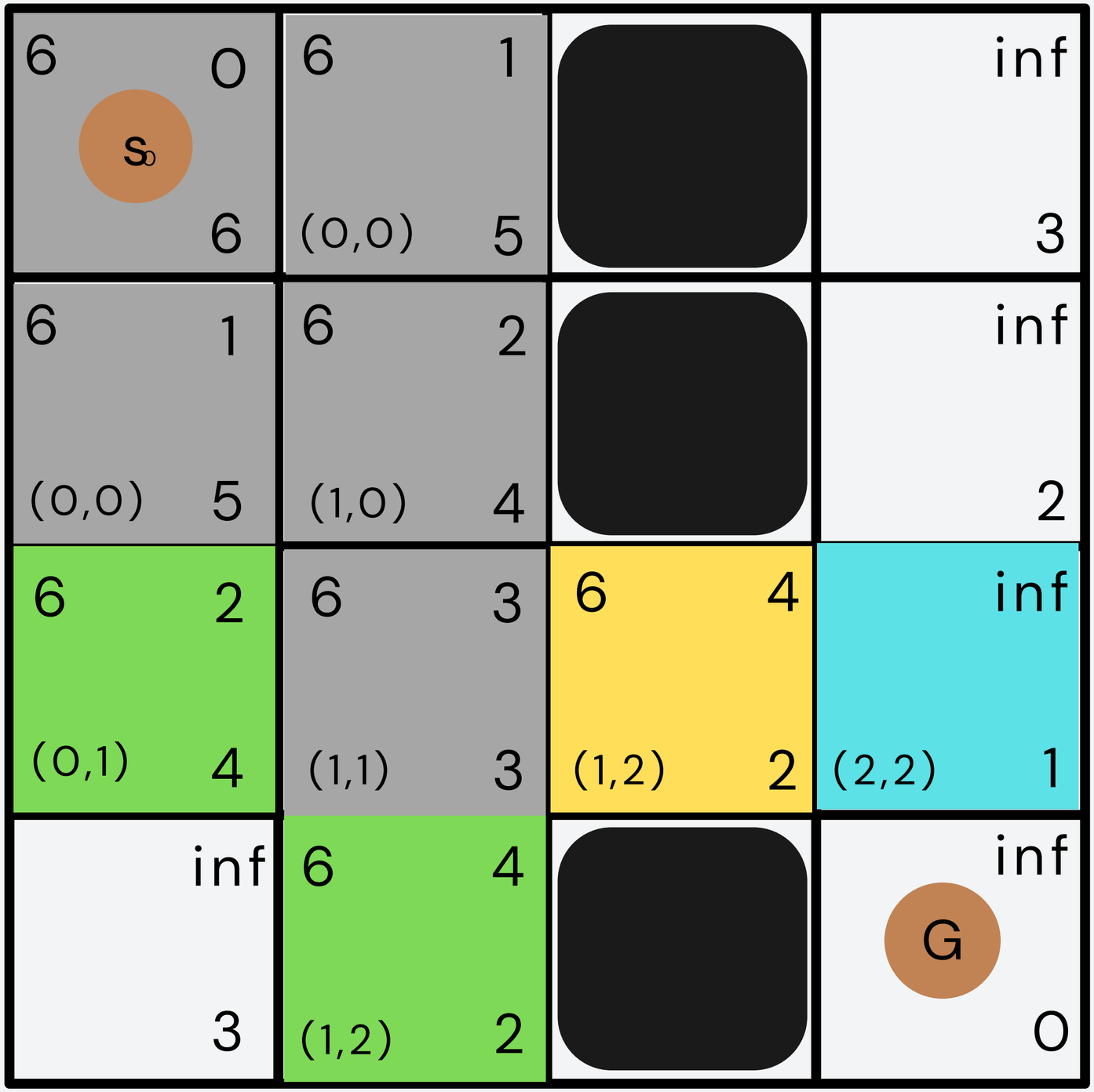
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



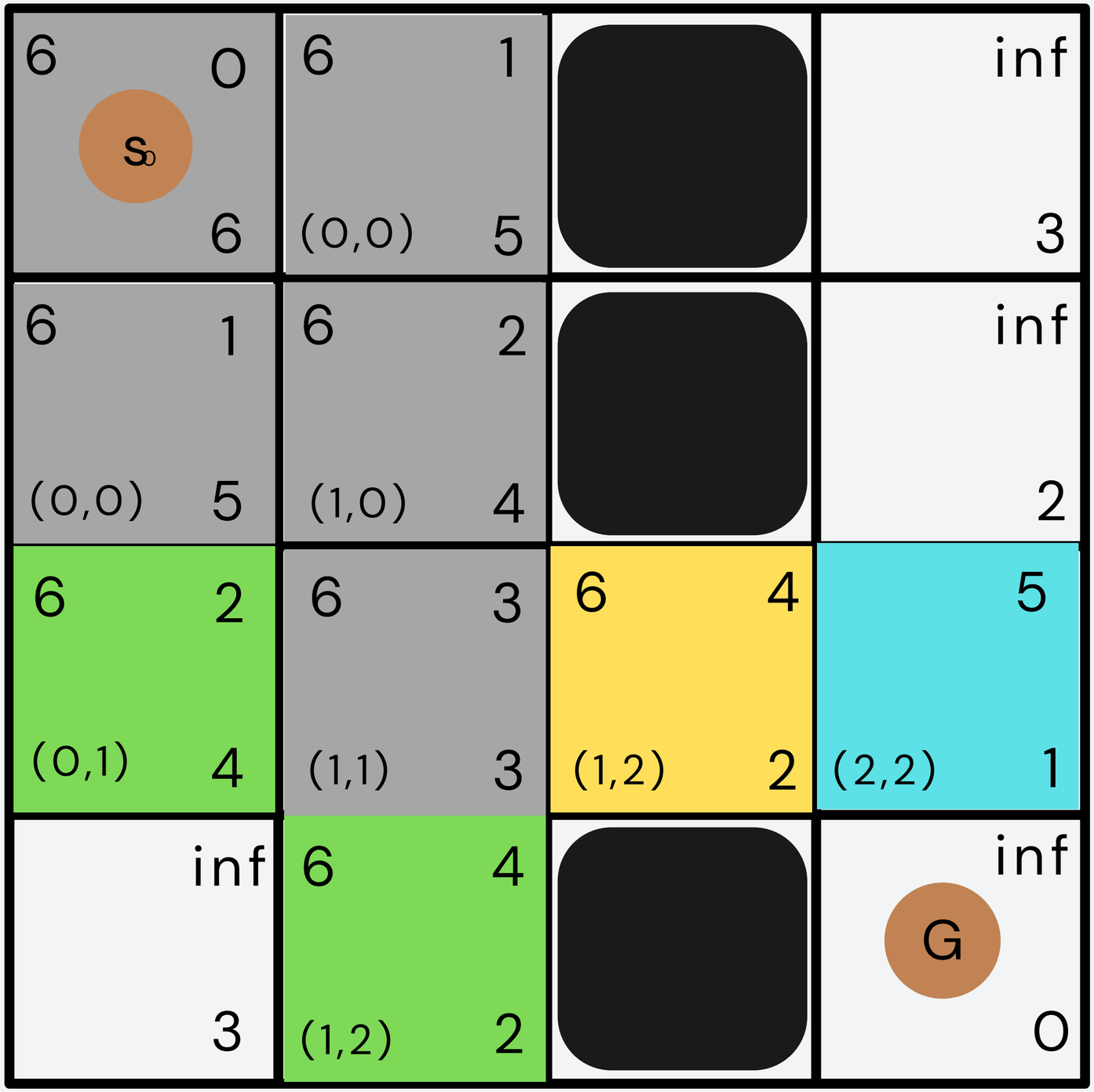
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



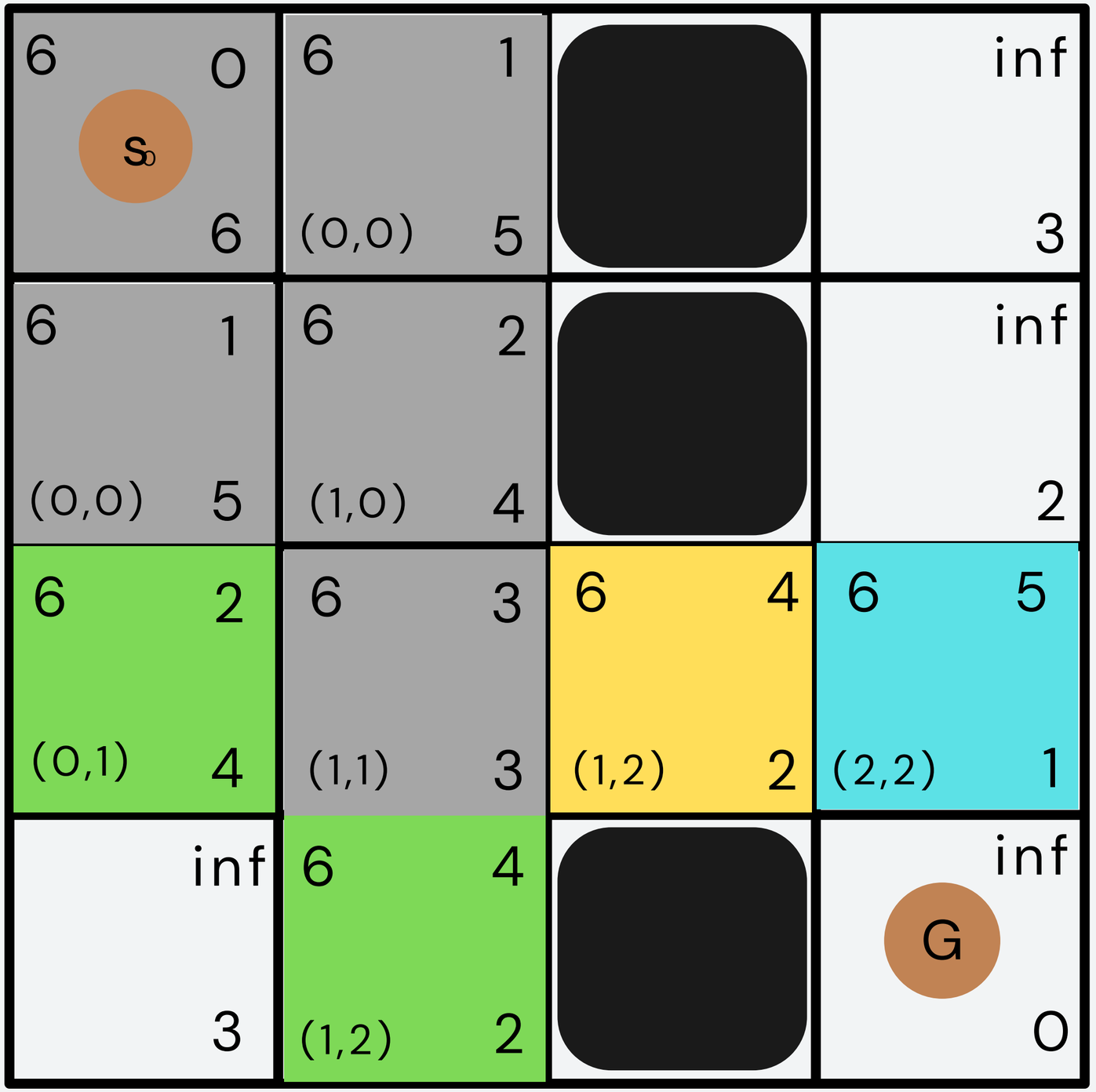
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



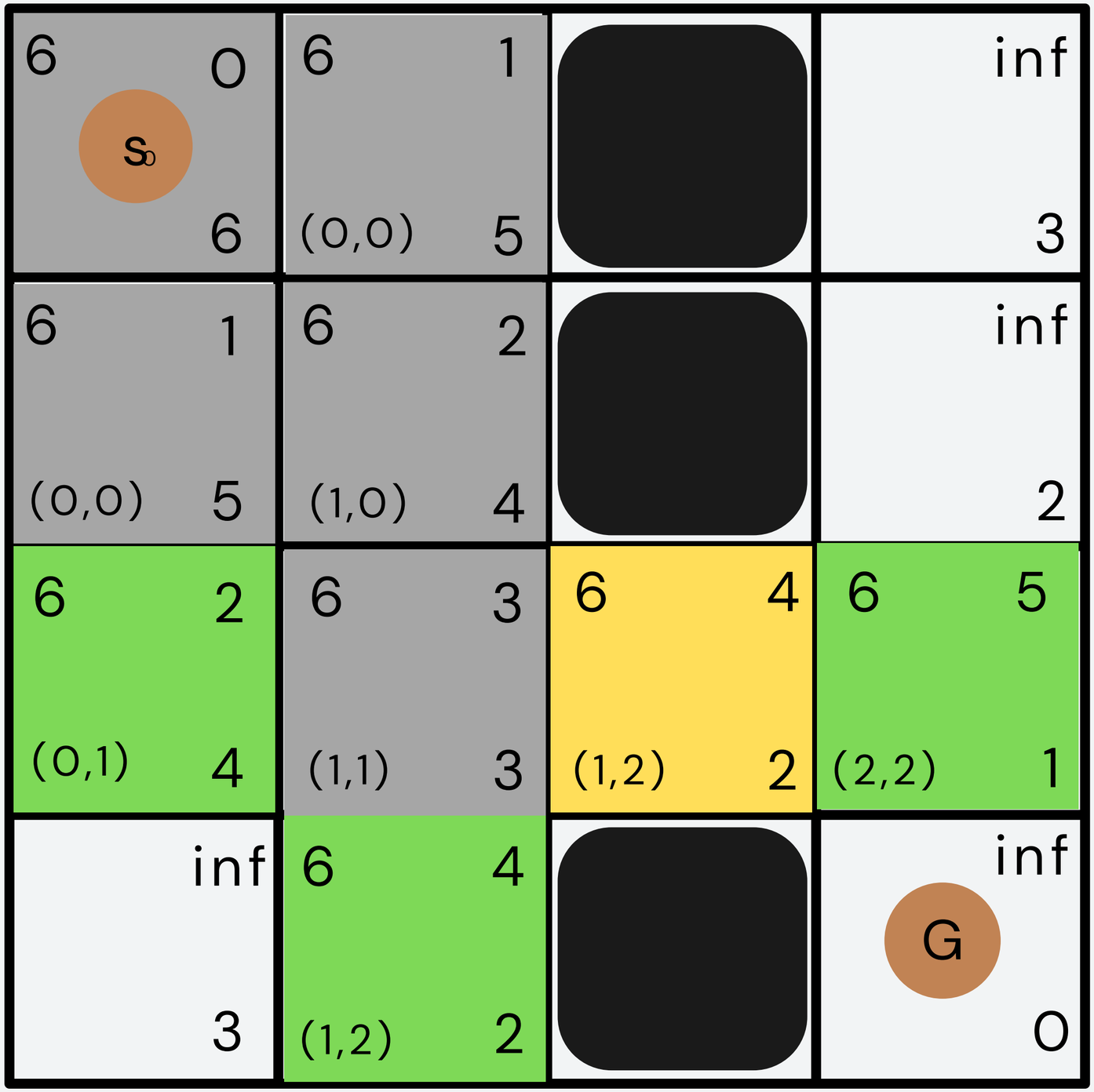
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



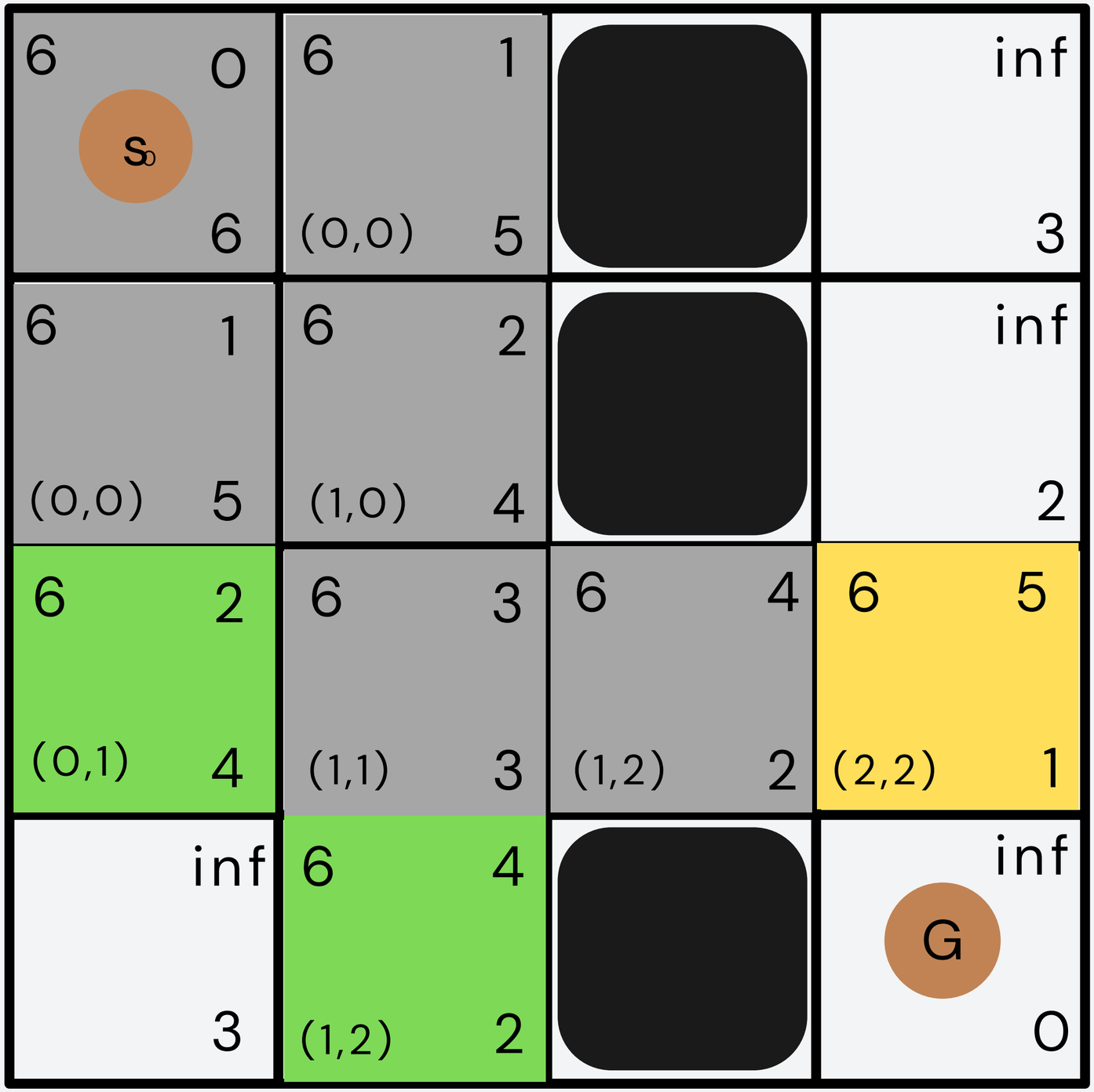
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



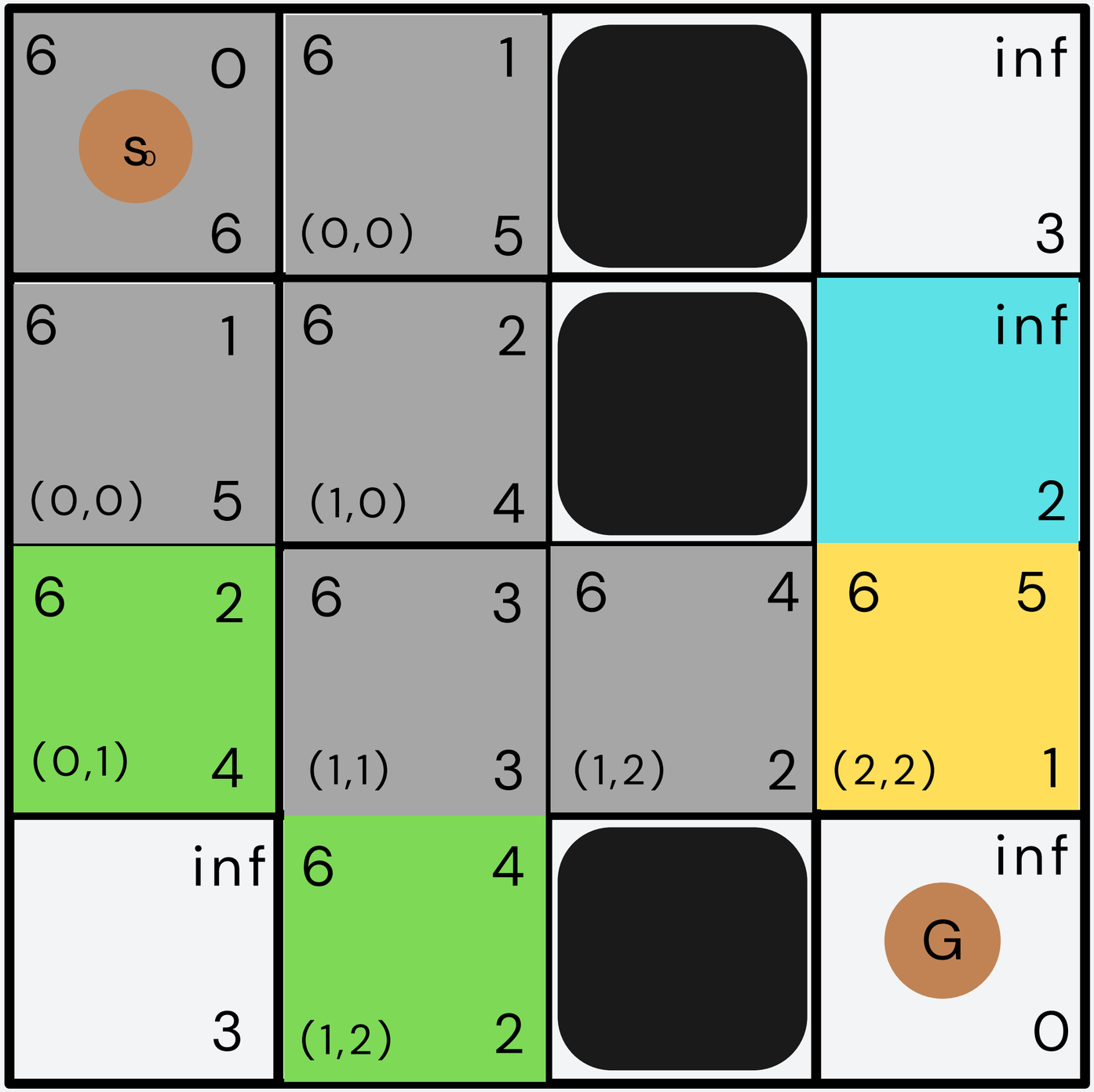
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



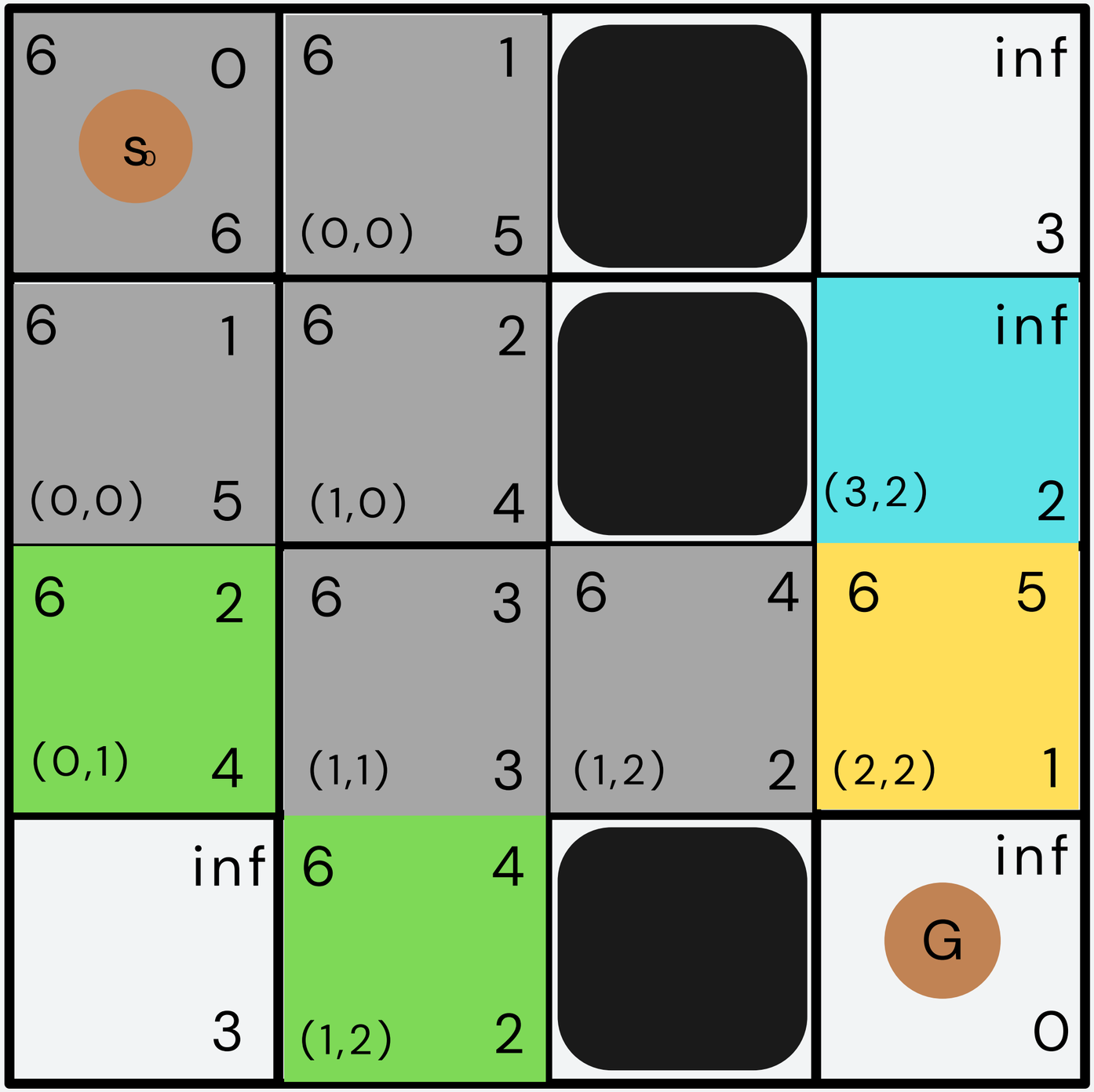
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



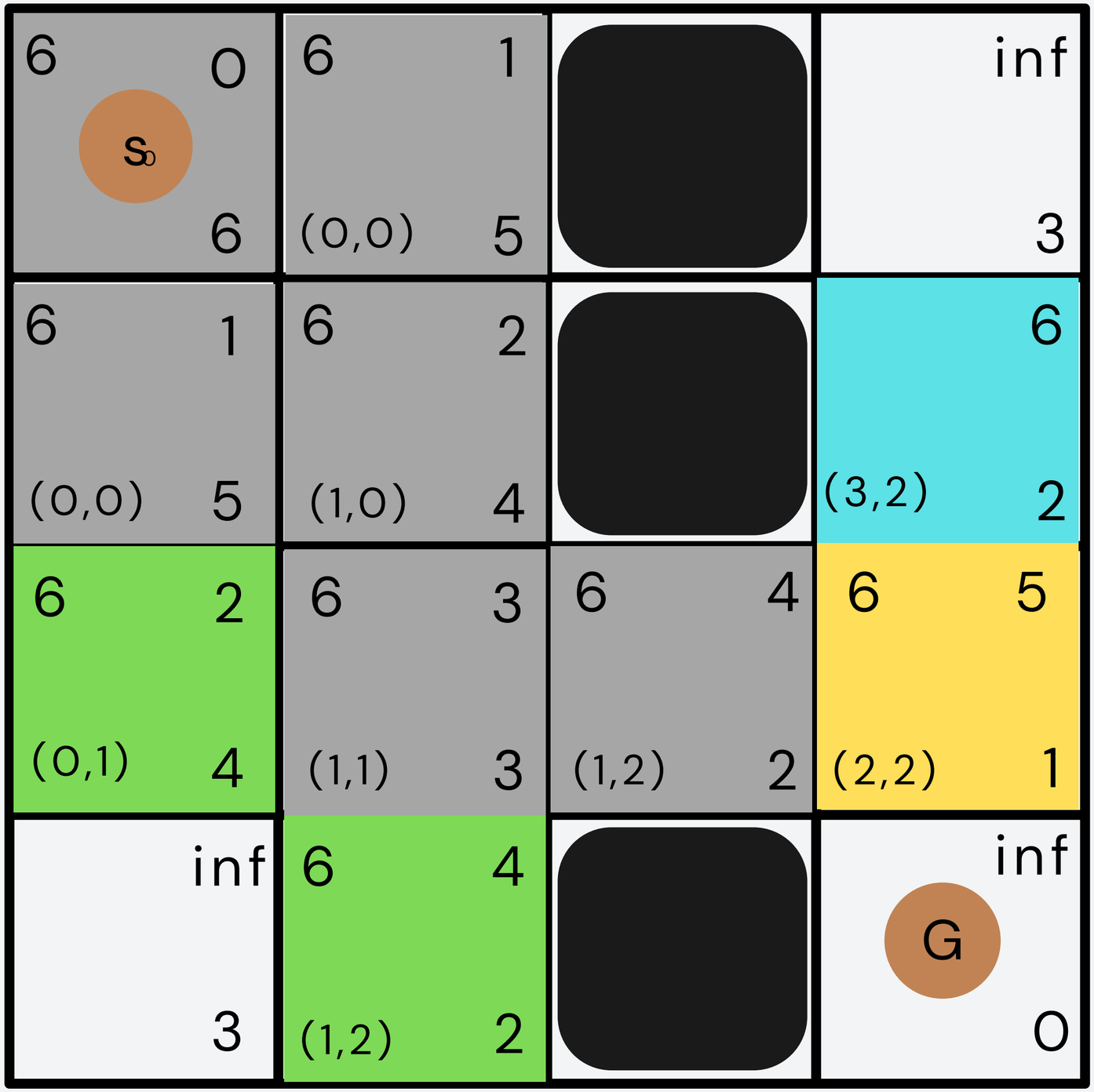
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



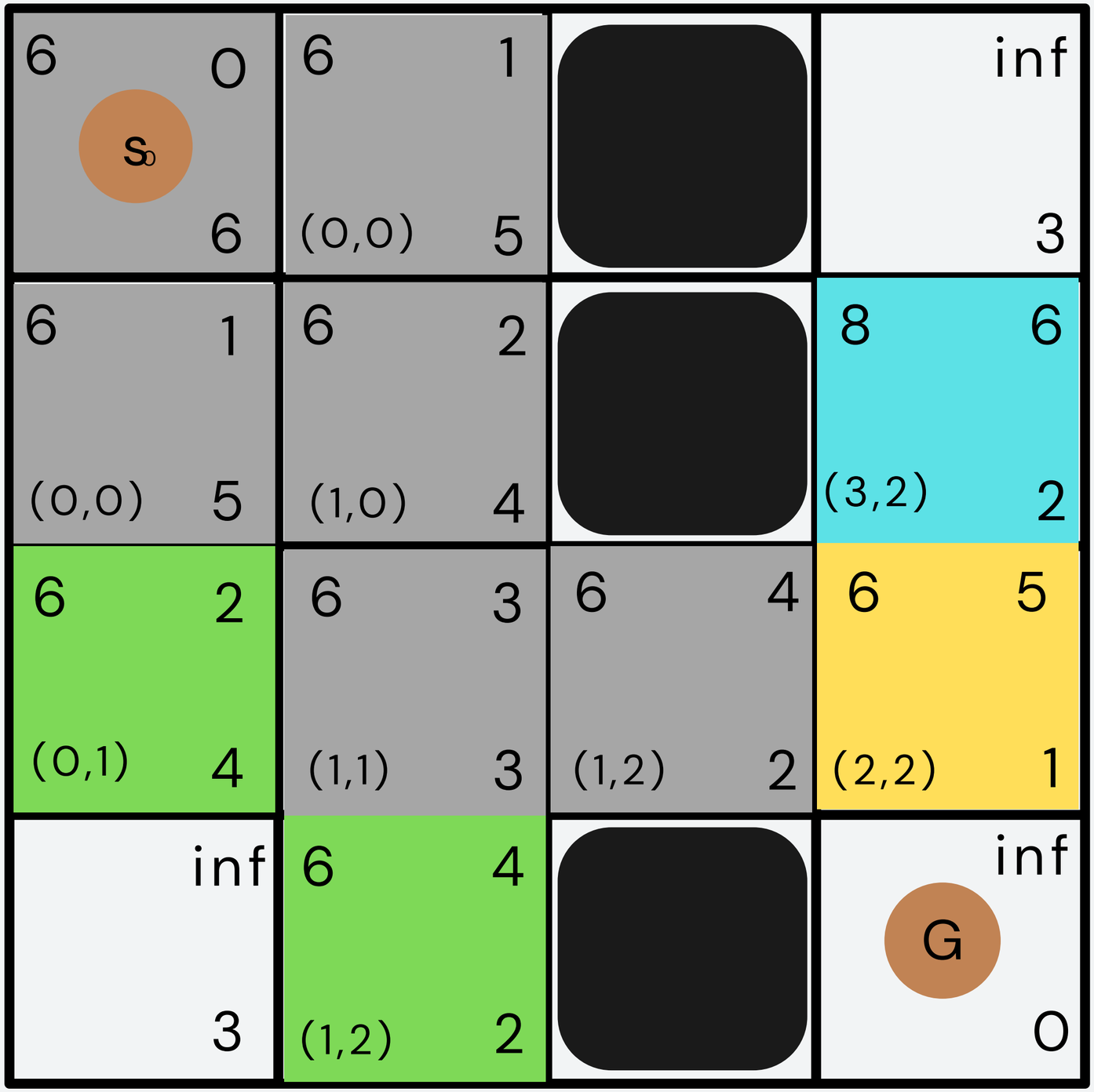
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



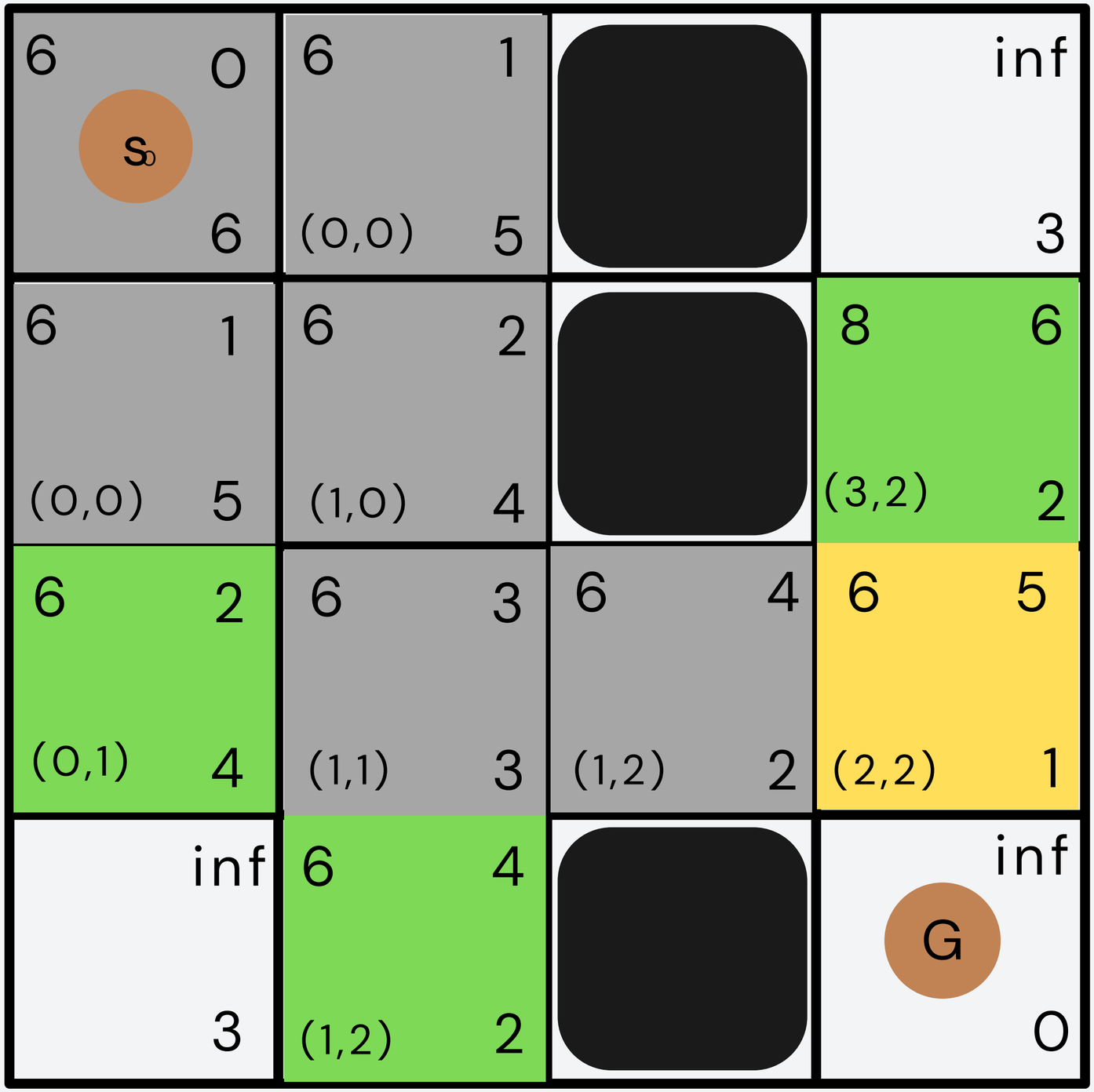
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



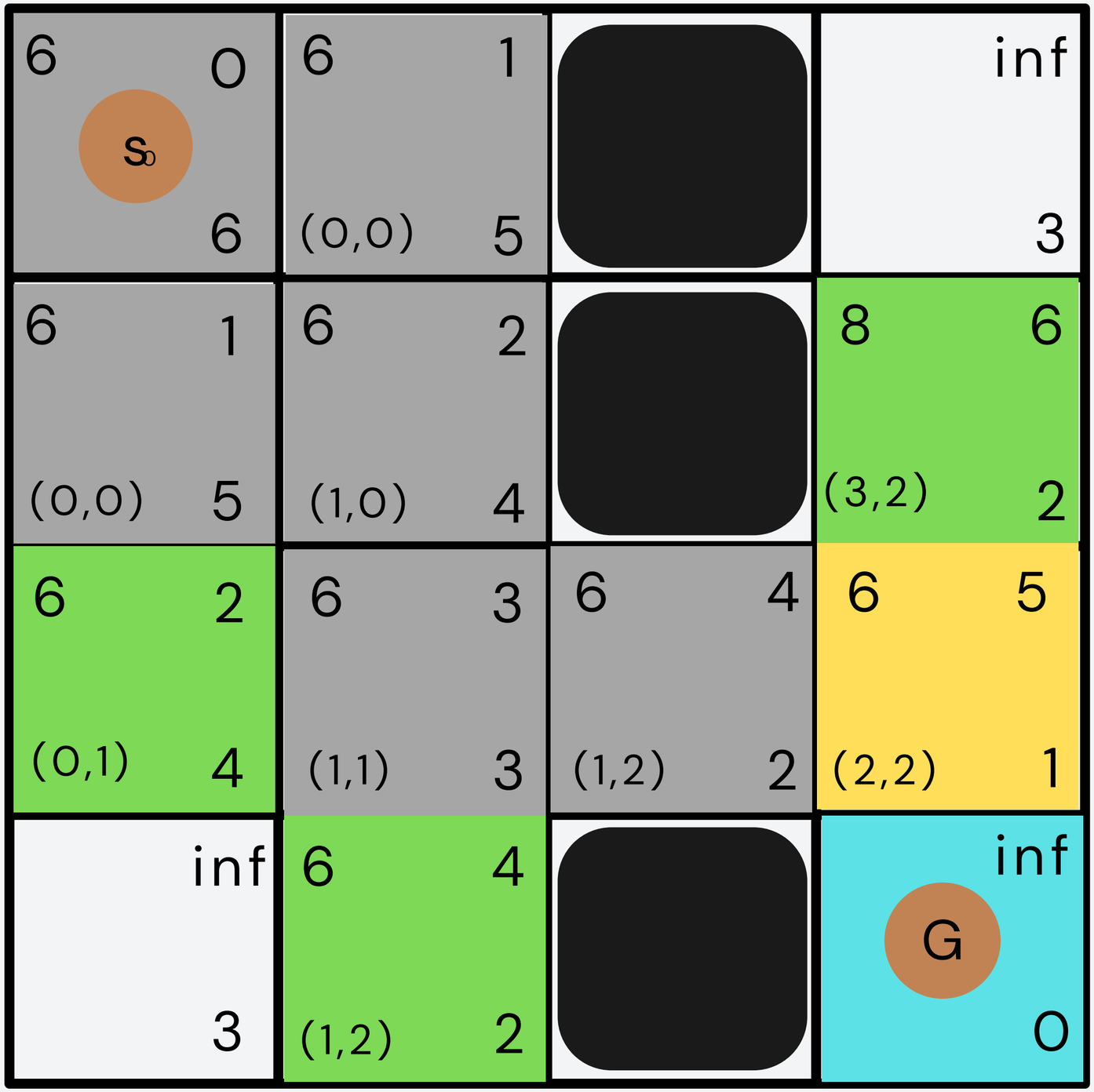
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



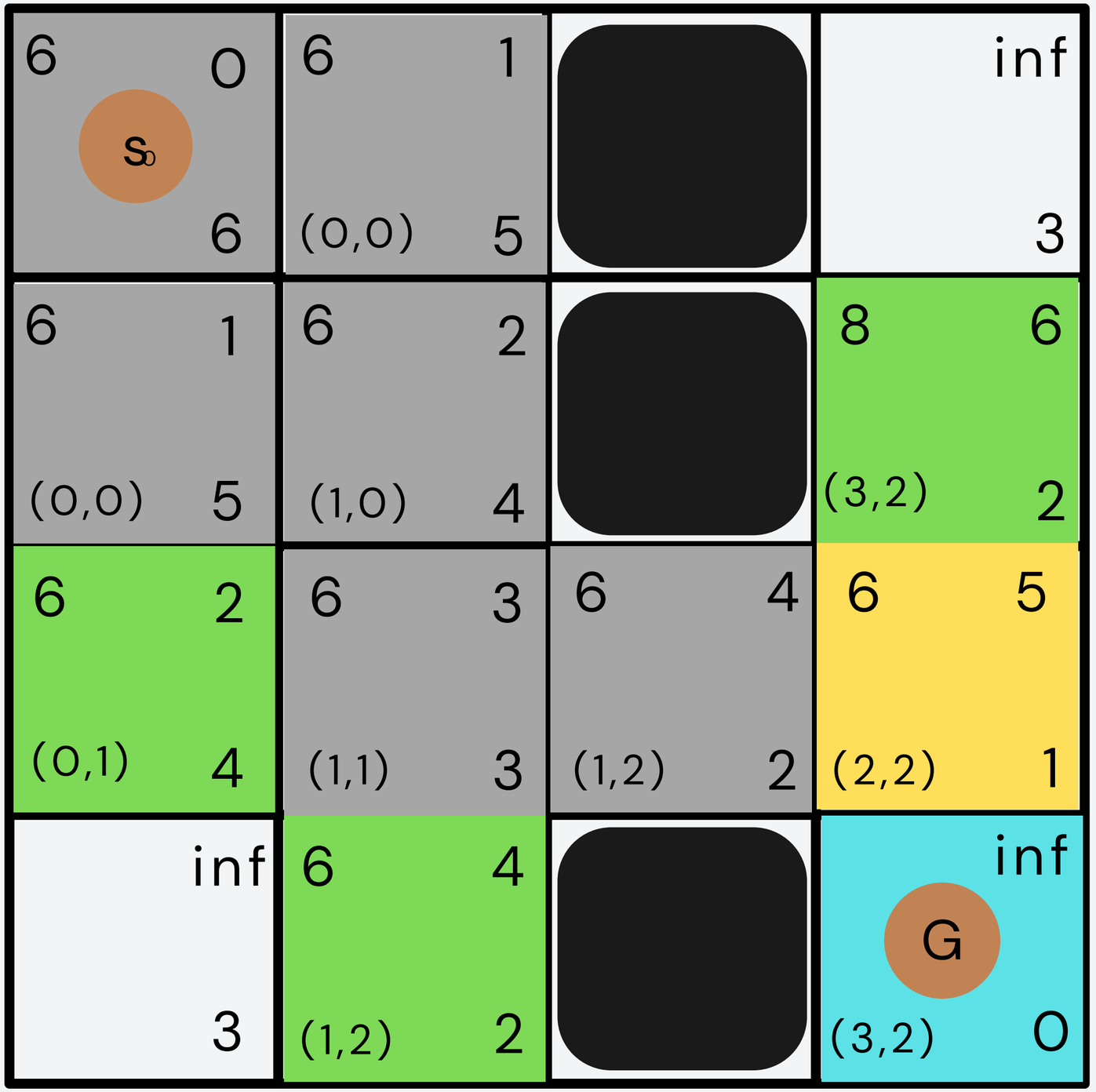
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



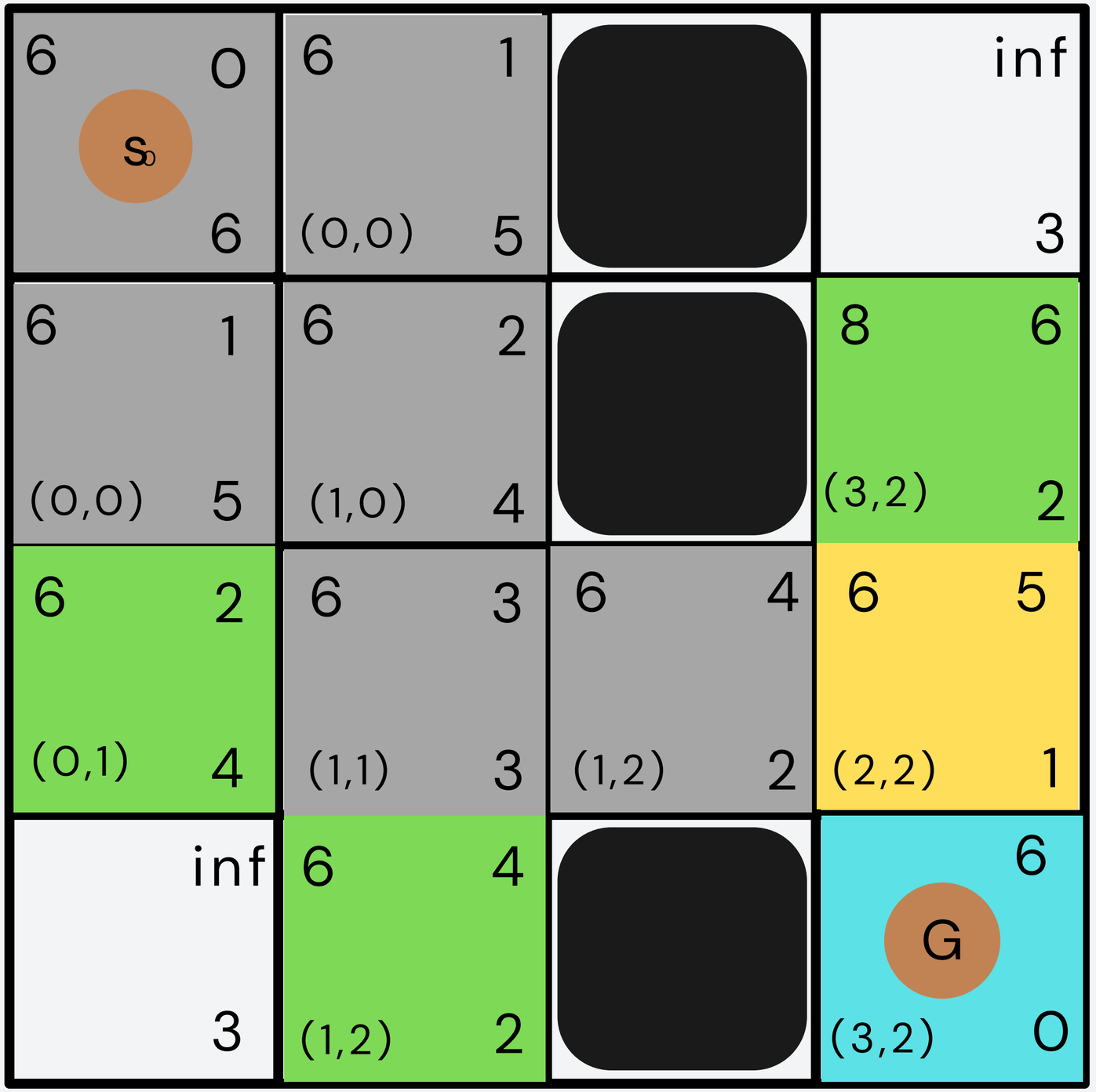
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



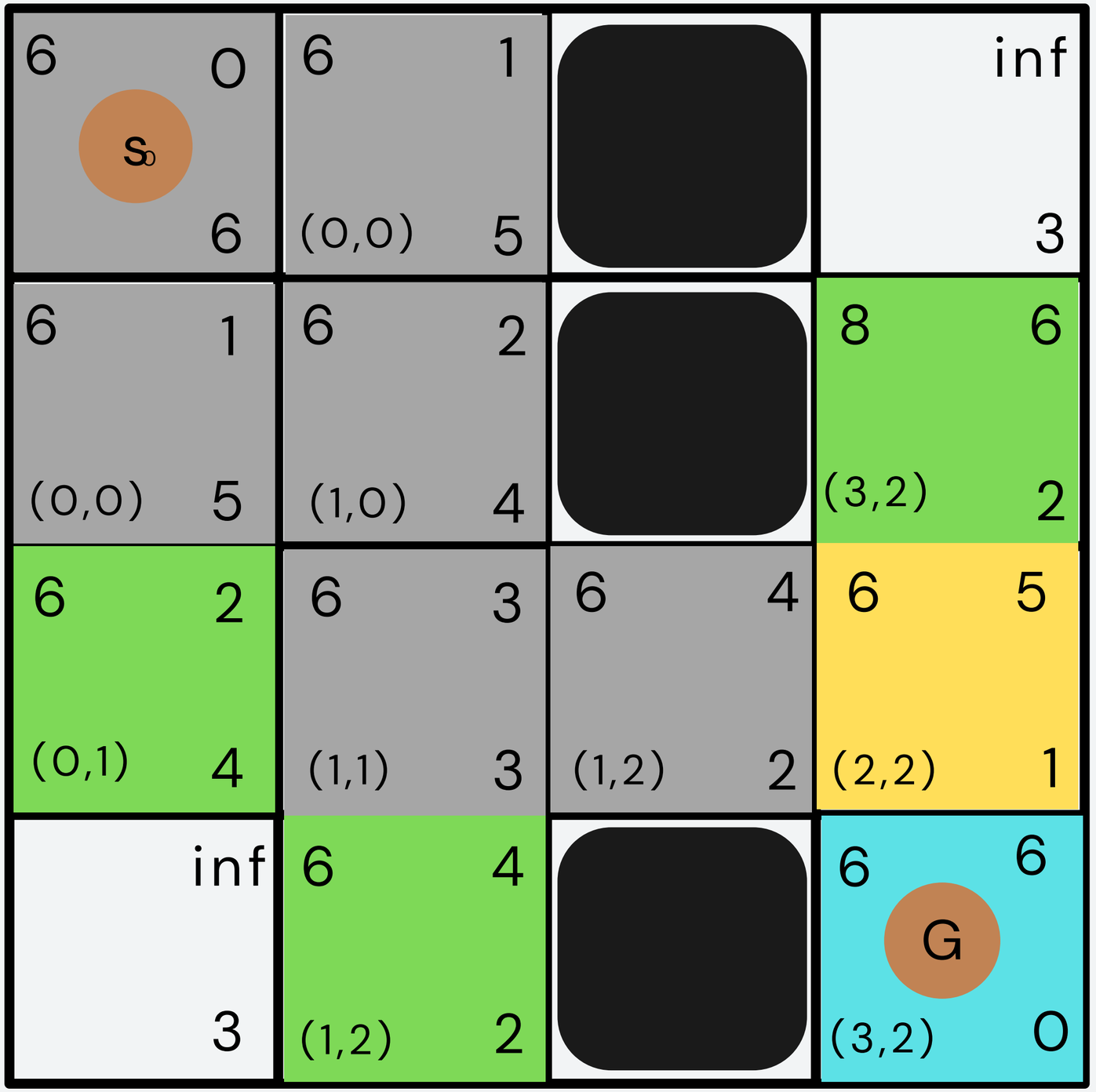
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



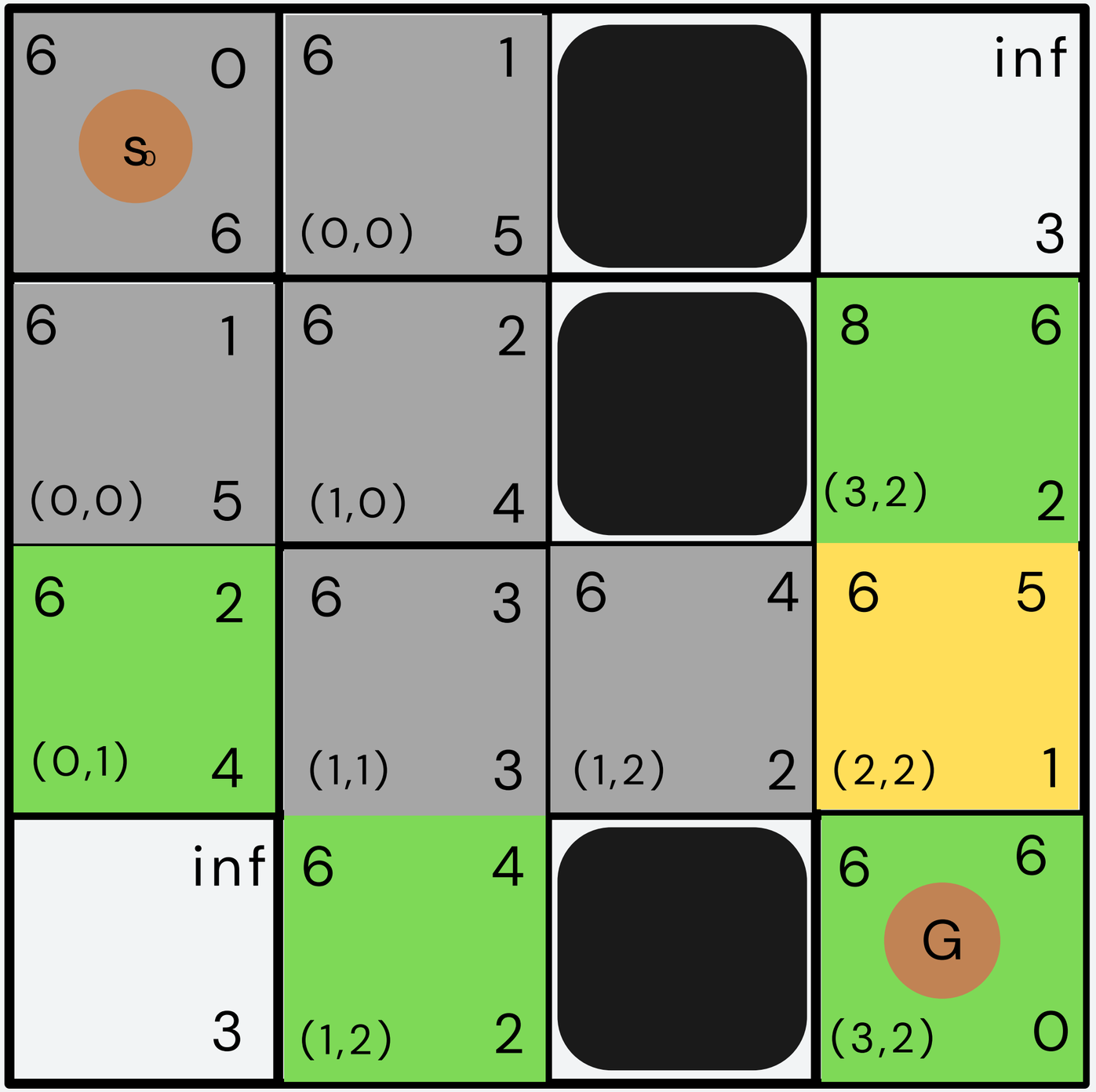
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



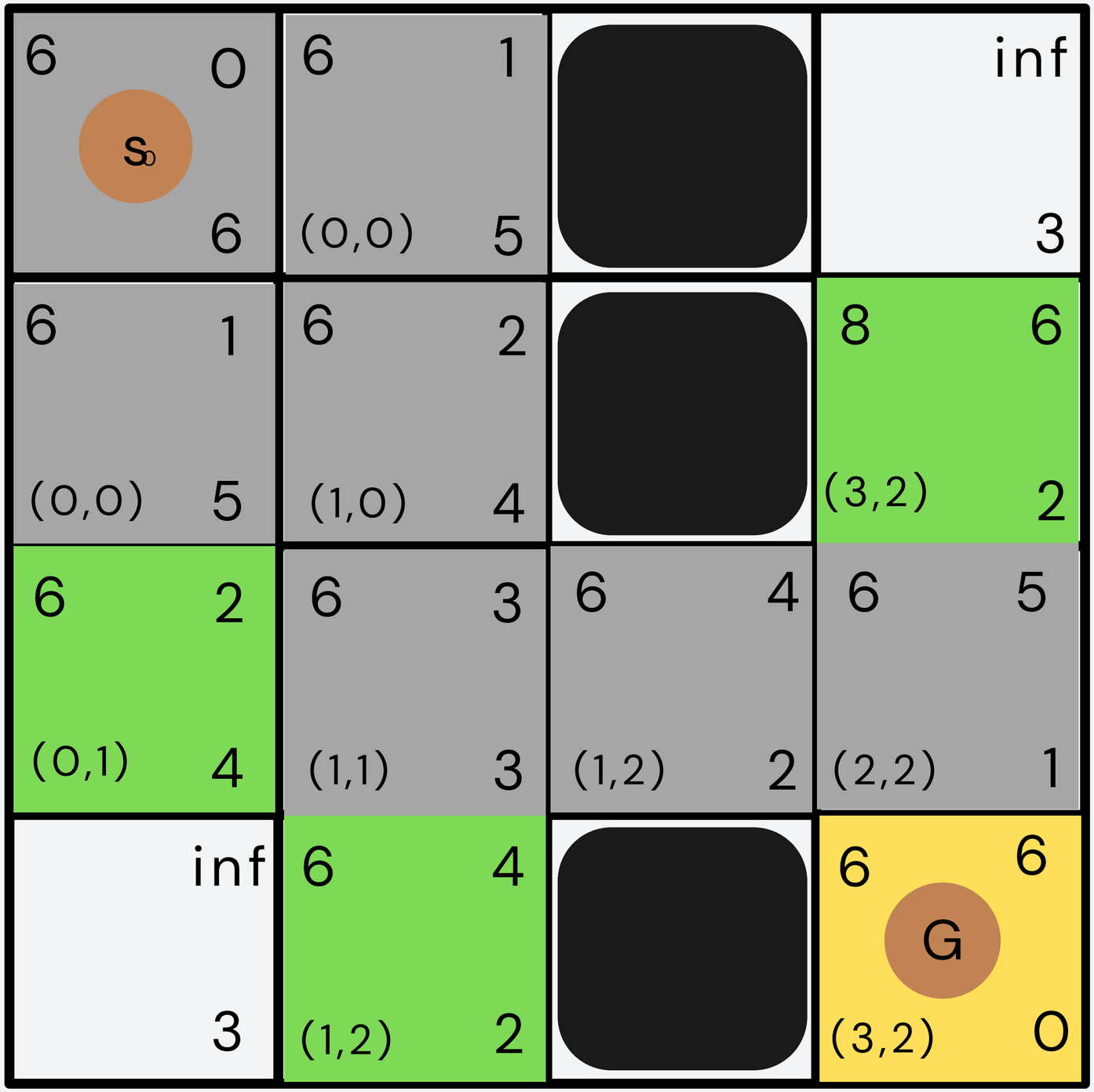
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
    1. $cost_v = g(u) + c(u, v)$
    2. **if** $cost_v \geq g(v)$ **return**
    3. $parent(v) \leftarrow u$
    4. $g(v) \leftarrow cost_v$
    5. $f(v) \leftarrow g(v) + h(v)$
    6. **if** $v \in Open$ **then** Reordenar $Open$
    7. **else** Insertar v en $Open$



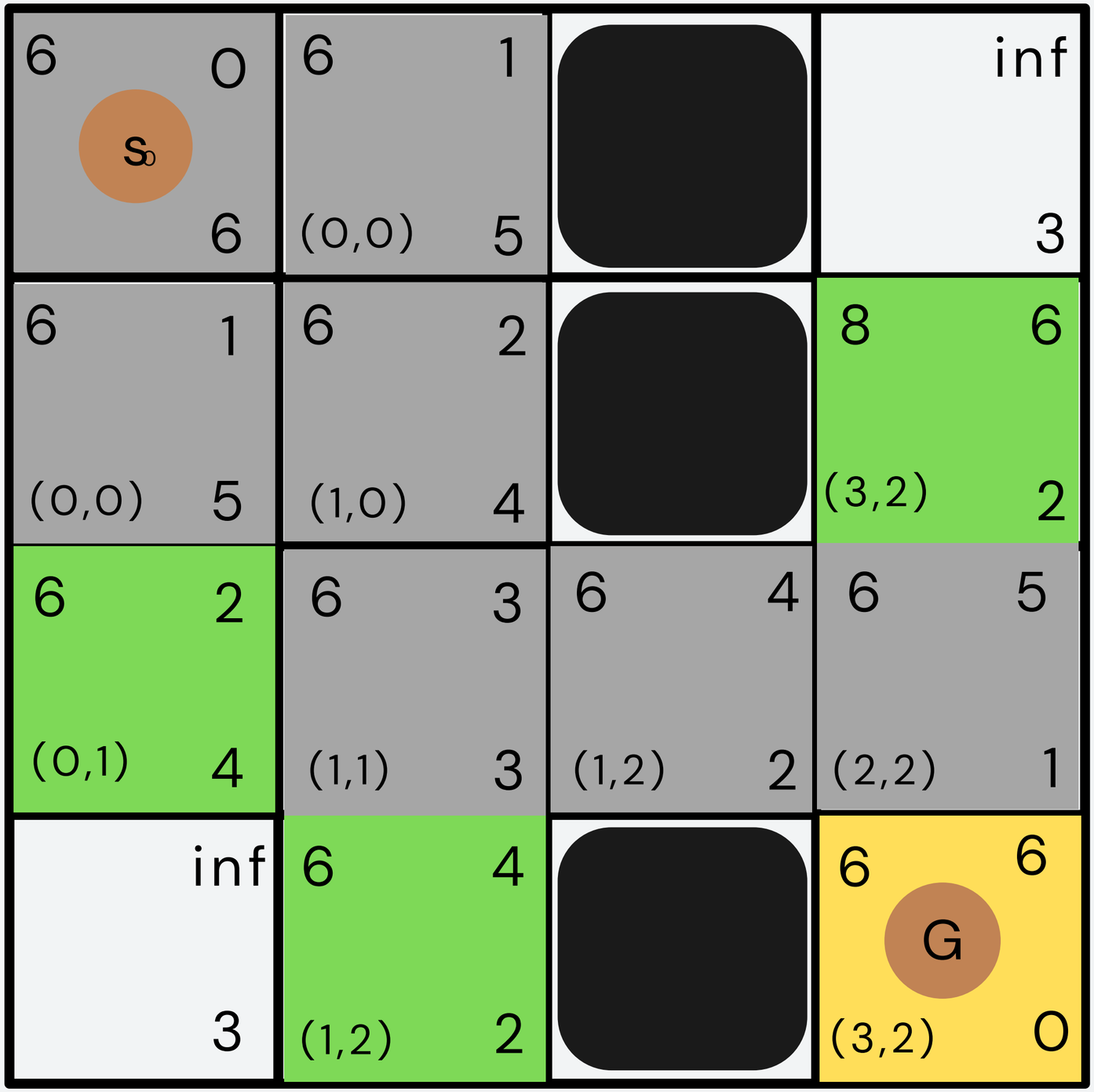
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



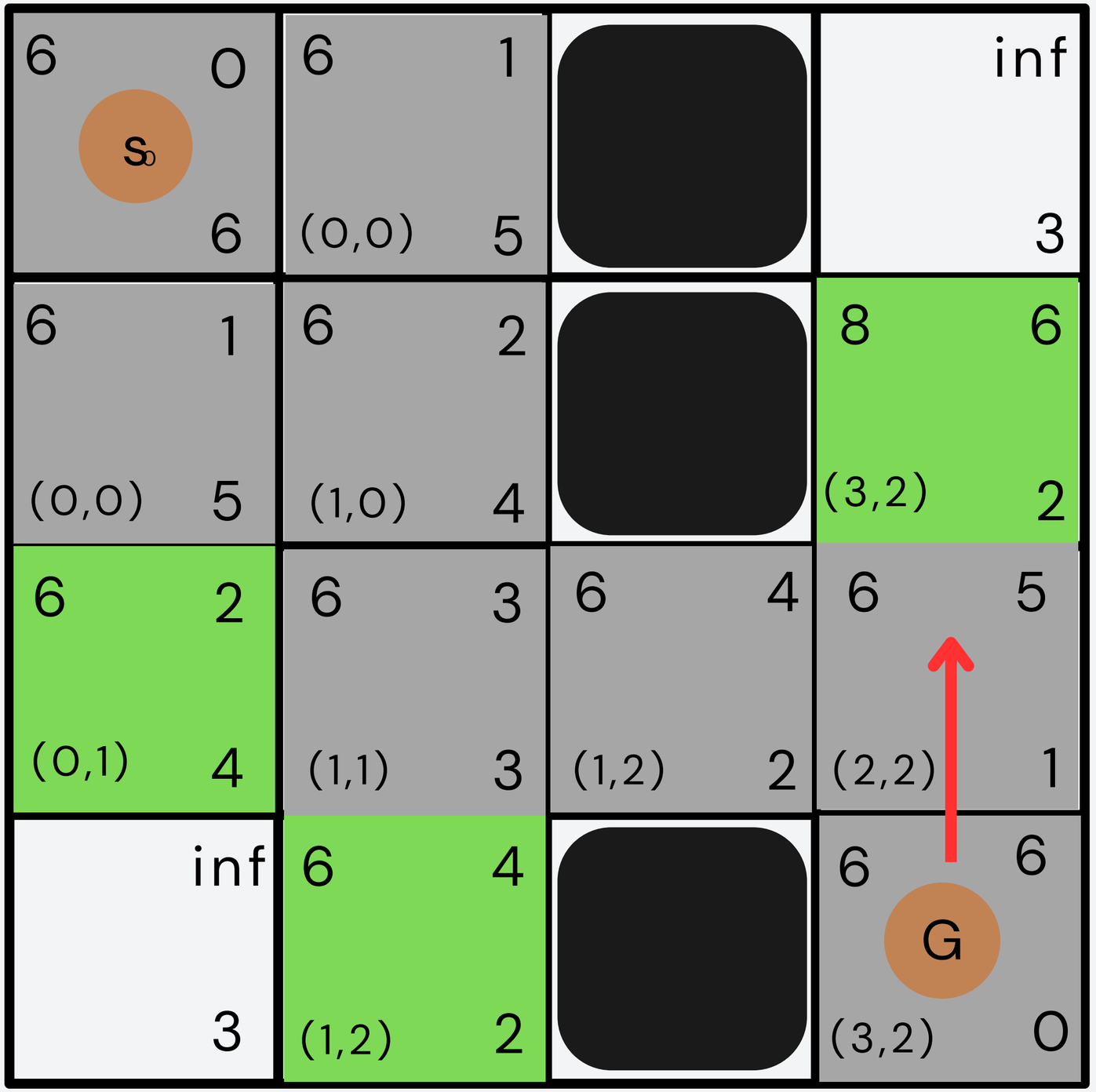
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



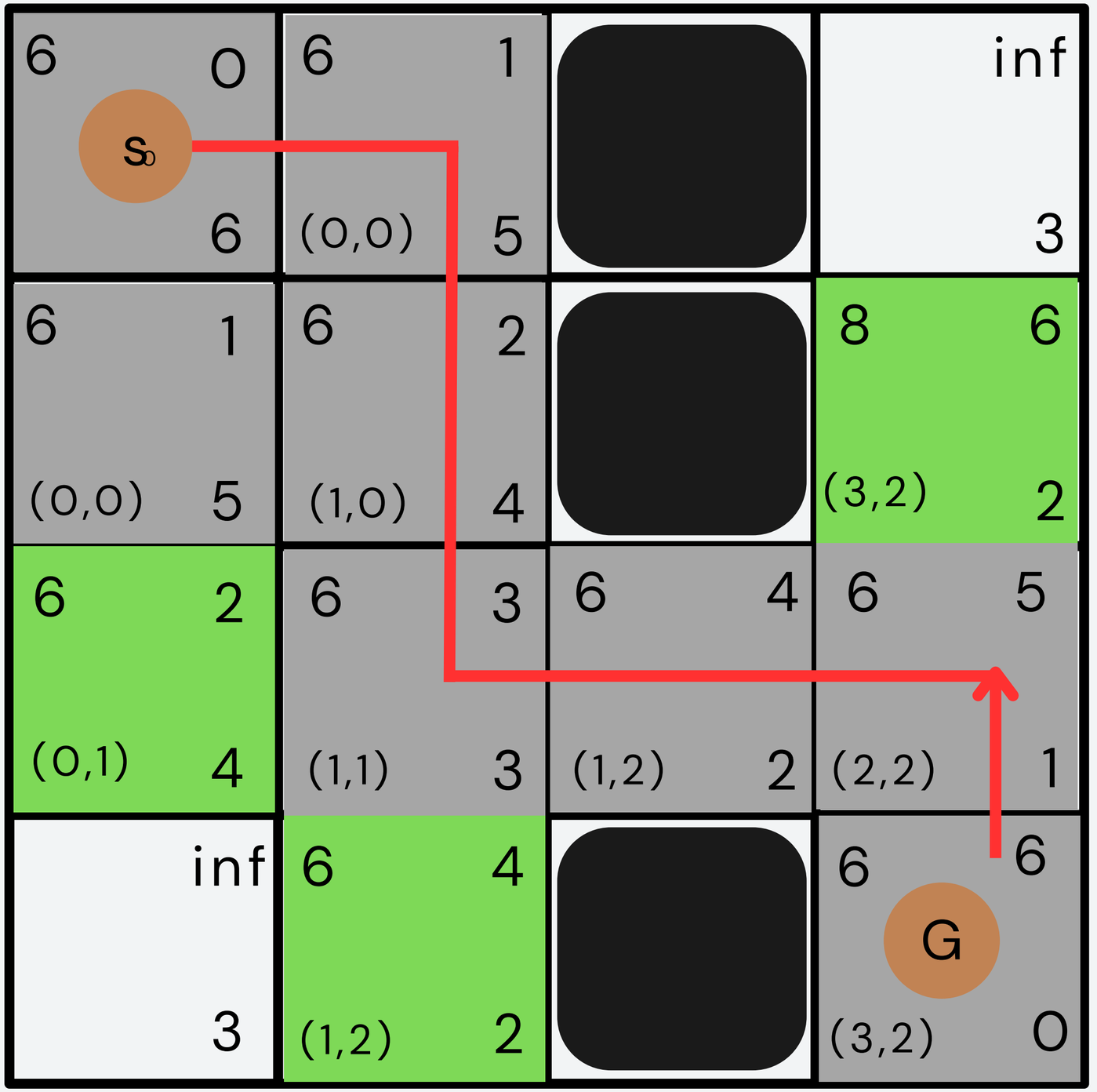
1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



1. **for each** $s \in \mathcal{S}$ **do** $g(s) \leftarrow \infty$
2. $Open \leftarrow \{s_0\}$
3. $g(s_0) \leftarrow 0;\ f(s_0) \leftarrow h(s_0)$
4. **while** $Open \neq \emptyset$
5. Extrae un u desde $Open$ con menor valor-f
6. **if** u es objetivo **return** u
7. **for each** $v \in Succ(u)$ **do**
   1. $cost_v = g(u) + c(u, v)$
   2. **if** $cost_v \geq g(v)$ **return**
   3. $parent(v) \leftarrow u$
   4. $g(v) \leftarrow cost_v$
   5. $f(v) \leftarrow g(v) + h(v)$
   6. **if** $v \in Open$ **then** Reordenar $Open$
   7. **else** Insertar v en $Open$



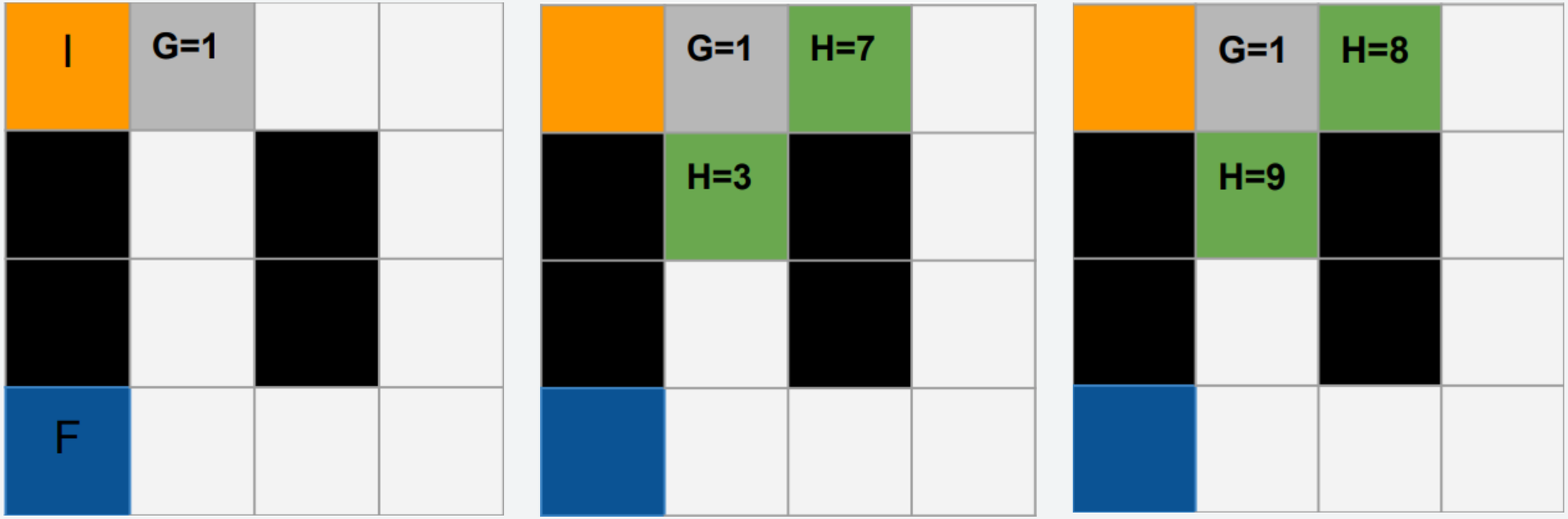
OPTIMALIDAD

ADMISIBILIDAD

Una heurística h(s) es admisible si no sobreestima el costo real para llegar al objetivo, es decir

$$h(s) \leq h^*(s)$$

ADMISIBILIDAD



CONSISTENCIA

Una heurística es consistente si el valor de la heurística en un nodo s debe ser menor o igual que la suma del valor de la heurística de sus vecinos y el costo de llegar a cada uno de ellos.

$$h(s) = 0, \ \forall s \text{ en } G$$

$$h(s) \leq c(s,s') + h(s'), \text{ para todo vecinos de } s.$$

¿Por qué A* es óptimo? - YouTube

https://www.youtube.com/watch?v=_41v4I5GTNc