

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)



	G	

	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
ŀ.	if (node.state in closed) continue
5.	add node to closed
5.	for c in expand(node)
<i>'</i> .	if c.state in closed: continue
3.	if c in open with $\leq =$ g: continue
).	else open.add(c)

Open List (Nodes)						Closed
S	G	Н	F	Α	Ρ	State
0,0	0	523	523			
	<b>S</b> 0,0	<b>S G</b> 0,0 0	S         G         H           0,0         0         523	S         G         H         F           0,0         0         523         523           0,0         0         523         523	S         G         H         F         A           0,0         0         523         523           0,0         0         523         523           0,0         0         523         523           0,0         0         523         523           0,0         0         523         523           0,0         0         523         523           0,0         0         523         523           0,0         0         523         523           0,0         0         523         523	S         G         H         F         A         P           0,0         0         523         523

<b>N</b> o <sub>G 0</sub>			
		G	

	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
ŀ.	if (node.state in closed) continue
5.	add node to closed
ō.	for c in expand(node)
<i>'</i> .	if c.state in closed: continue
3.	if c in open with $\leq =$ g: continue
).	else open.add(c)

Open List (Nodes)							Closed
ID	S	G	Н	F	Α	Ρ	State
0	0,0	0	523	523			

No g 0		
		G

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Open List (Nodes)							Closed
ID	S	G	Н	F	Α	Ρ	State
							0,0

No g 0	<b>C</b> G 100	
		G

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
1.	if (node.state in closed) continue
5.	add node to closed
5.	for c in expand(node)
7.	if c.state in closed: continue
3.	if c in open with $\leq =$ g: continue
Э.	else open.add(c)

Ope	en Lie	st (No	odes)				Closed
ID	S	G	Н	F	Α	Ρ	State
							0,0

No g 0	<b>C</b> G 100	
		G

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lis	st (No	des)				Closed
ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lie	st (No	des)				Closed
ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lie	st (No	des)				Closed
ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
2	1,1	141	382	523	1,1	0	



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lis	st (No	des)				Closed
ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
2	1,1	141	382	523	1,1	0	



ID	S	G	н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
2	1,1	141	382	523	1,1	0	
3	0,1	100	423	523	0,1	0	

Closed



Open List (Nodes)

[D	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
2	1,1	141	382	523	1,1	0	
3	0,1	100	423	523	0,1	0	

while (!open.empty) 1. node = remove min f from open 2. if (node.state is goal) return path 3. if (node.state in closed) continue 4. add node to closed 5. for c in expand(node) 6. if c.state in closed: continue 7. if c in open with  $\leq g$ : continue 8. else open.add(c) 9.

Closed



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Open	List	(Nodes)
------	------	---------

ID	S	G	н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
2	1,1	141	382	523	1,1	0	
3	0,1	100	423	523	0,1	0	



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lis	st (No	des)				Closed
ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
3	0,1	100	423	523	0,1	0	1, 1



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lie	st (No	des)				Closed
ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
3	0,1	100	423	523	0,1	0	1, 1



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with $\leq =$ g: continue
9.	else open.add(c)

Ope	en Lie	st (No	des)				Closed
ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
3	0,1	100	423	523	0,1	0	1, 1



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lie	st (No	des)				Closed
ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
3	0,1	100	423	523	0,1	0	1, 1



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lis	st (No	des)				Closed
ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
3	0,1	100	423	523	0,1	0	1, 1





ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
3	0,1	100	423	523	0,1	0	1, 1
4	2,0	282	441	723	1,-1	2	



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with $\leq =$ g: continue
9.	else open.add(c)



ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
3	0,1	100	423	523	0,1	0	1, 1
4	2,0	282	441	723	1,-1	2	



	nc	COC	
		<b>u</b>	

ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
3	0,1	100	423	523	0,1	0	1, 1
4	2,0	282	441	723	1,-1	2	
5	1,2	241	282	523	0,1	2	

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)



Clos	sed	

ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
3	0,1	100	423	523	0,1	0	1,1
4	2,0	282	441	723	1,-1	2	
5	1,2	241	282	523	0,1	2	



ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
3	0,1	100	423	523	0,1	0	1, 1
4	2,0	282	441	723	1,-1	2	
5	1,2	241	282	523	0,1	2	
6	0,2	282	382	664	-1,1	2	



	~		4
	05	<b>5</b> 80	

ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
3	0,1	100	423	523	0,1	0	1, 1
4	2,0	282	441	723	1,-1	2	
5	1,2	241	282	523	0,1	2	
6	0,2	282	382	664	-1,1	2	



	~		4
	05	<b>5</b> 80	

ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
3	0,1	100	423	523	0,1	0	1, 1
4	2,0	282	441	723	1,-1	2	
5	1,2	241	282	523	0,1	2	
6	0,2	282	382	664	-1,1	2	



C	ACC	bd
	USE	zu

ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
3	0,1	100	423	523	0,1	0	1, 1
4	2,0	282	441	723	1,-1	2	
5	1,2	241	282	523	0,1	2	
6	0,2	282	382	664	-1,1	2	



Open List (Nodes)

ID

1

3

4

5

6

CII	LR	Closed					
9	5	G	Н	F	Α	Ρ	State
1	,0	100	482	582	1,0	0	0,0
0	,1	100	423	523	0,1	0	1, 1
2,	,0	282	441	723	1,-1	2	
1	,2	241	282	523	0,1	2	
0	,2	282	382	664	-1,1	2	





ID	S	G	н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
5	1,2	241	282	523	0,1	2	0,1
6	0,2	282	382	664	-1,1	2	

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)





ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
5	1,2	241	282	523	0,1	2	0,1
6	0,2	282	382	664	-1,1	2	





ID	S	G	н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
5	1,2	241	282	523	0,1	2	0,1
6	0,2	282	382	664	-1,1	2	

while (!open.empty) 1. node = remove min f from open 2. if (node.state is goal) return path 3. if (node.state in closed) continue 4. add node to closed 5. for c in expand(node) 6. if c.state in closed: continue 7. if c in open with  $\leq g$ : continue 8. else open.add(c) 9.





ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
5	1,2	241	282	523	0,1	2	0,1
6	0,2	282	382	664	-1,1	2	

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)





ID	S	G	н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
5	1,2	241	282	523	0,1	2	0,1
6	0,2	282	382	664	-1,1	2	

while (!open.empty) 1. node = remove min f from open 2. if (node.state is goal) return path 3. if (node.state in closed) continue 4. add node to closed 5. for c in expand(node) 6. if c.state in closed: continue 7. if c in open with  $\leq g$ : continue 8. else open.add(c) 9.





ID	S	G	н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
5	1,2	241	282	523	0,1	2	0,1
6	0,2	282	382	664	-1,1	2	

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with $\leq =$ g: continue
9.	else open.add(c)





ID	S	G	н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
5	1,2	241	282	523	0,1	2	0,1
6	0,2	282	382	664	-1,1	2	

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with $\leq =$ g: continue
9.	else open.add(c)





ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
5	1,2	241	282	523	0,1	2	0,1
6	0,2	282	382	664	-1,1	2	
7	0,2	200	382	582	0,1	3	



Open List (Nodes)



ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
5	1,2	241	282	523	0,1	2	0,1
6	0,2	282	382	664	-1,1	2	
7	0,2	200	382	582	0,1	3	

Important: Node 6 Will Never Expand Remove (if efficient)





ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
5	1,2	241	282	523	0,1	2	0,1
6	0,2	282	382	664	-1,1	2	
7	0,2	200	382	582	0,1	3	




ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
5	1,2	241	282	523	0,1	2	0,1
7	0,2	200	382	582	0,1	3	

while (!open.empty) 1. node = remove min f from open 2. if (node.state is goal) return path 3. if (node.state in closed) continue 4. add node to closed 5. for c in expand(node) 6. if c.state in closed: continue 7. if c in open with  $\leq g$ : continue 8. else open.add(c) 9.



$\mathbf{C}$	lacad	
L	IUSEU	

ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1,1
5	1,2	241	282	523	0,1	2	0,1
7	0,2	200	382	582	0,1	3	

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)



while (!open.empty) 1. node = remove min f from open 2. if (node.state is goal) return path 3. if (node.state in closed) continue 4. add node to closed 5. for c in expand(node) 6. if c.state in closed: continue 7. if c in open with  $\leq g$ : continue 8. else open.add(c) 9.



ID	S	G	н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
5	1,2	241	282	523	0,1	2	0,1
7	0,2	200	382	582	0,1	3	



while (!open.empty) 1. node = remove min f from open 2. if (node.state is goal) return path 3. if (node.state in closed) continue 4. add node to closed 5. for c in expand(node) 6. if c.state in closed: continue 7. if c in open with  $\leq g$ : continue 8. else open.add(c) 9.



ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
7	0,2	200	382	582	0,1	3	0,1
							1, 2



while (!open.empty) 1. node = remove min f from open 2. if (node.state is goal) return path 3. if (node.state in closed) continue 4. add node to closed 5. for c in expand(node) 6. if c.state in closed: continue 7. if c in open with  $\leq g$ : continue 8. else open.add(c) 9.



ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
7	0,2	200	382	582	0,1	3	0,1
							1, 2





ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
7	0,2	200	382	582	0,1	3	0,1
							1, 2

while (!open.empty) 1. node = remove min f from open 2. if (node.state is goal) return path 3. if (node.state in closed) continue 4. add node to closed 5. for c in expand(node) 6. if c.state in closed: continue 7. if c in open with  $\leq g$ : continue 8.

9. else open.add(c)





ID	S	G	н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
7	0,2	200	382	582	0,1	3	0,1
							1, 2

while (!open.empty) 1. node = remove min f from open 2. if (node.state is goal) return path 3. if (node.state in closed) continue 4. add node to closed 5. for c in expand(node) 6. if c.state in closed: continue 7. if c in open with  $\leq g$ : continue 8.

9. else open.add(c)



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)



ID	S	G	Н	F	Α	Ρ	State
1	1,0	100	482	582	1,0	0	0,0
4	2,0	282	441	723	1,-1	2	1, 1
7	0,2	200	382	582	0,1	3	0,1
							1, 2



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Open List (Nodes)							Closed
ID	S	G	Н	F	Α	Ρ	State
4	2,0	282	441	723	1,-1	2	0,0
7	0,2	200	382	582	0,1	3	1, 1
							0,1
							1, 2
							1,0



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Open List (Nodes)					Closed		
ID	S	G	Н	F	Α	Ρ	State
4	2,0	282	441	723	1,-1	2	0,0
7	0,2	200	382	582	0,1	3	1, 1
							0,1
							1, 2
							1,0



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)



ID	S	G	н	F	Α	Ρ	State
4	2,0	282	441	723	1,-1	2	0,0
7	0,2	200	382	582	0,1	3	1, 1
8	2,0	200	441	641	1,0	1	0,1
							1, 2
							1,0



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with $\leq =$ g: continue
9.	else open.add(c)



ID	S	G	н	F	Α	Ρ	State
8	2,0	200	441	641	1,0	1	0,0
7	0,2	200	382	582	0,1	3	1, 1
							0,1
							1, 2
							1,0



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)



ID	S	G	Н	F	Α	Ρ	State
8	2,0	200	441	641	1,0	1	0,0
7	0,2	200	382	582	0,1	3	1, 1
							0,1
							1, 2
							1,0



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with $\leq =$ g: continue
9.	else open.add(c)

Open List (Nodes)						Closed	
ID	S	G	Н	F	Α	Ρ	State
8	2,0	200	441	641	1,0	1	0,0
7	0,2	200	382	582	0,1	3	1, 1
							0,1
							1, 2
							1,0



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with $\leq =$ g: continue
9.	else open.add(c)

Ope	en Lis	st (No	odes)				Closed
ID	S	G	Н	F	Α	Ρ	State
8	2,0	200	441	641	1,0	1	0,0
							1, 1
							0,1
							1, 2
							1,0
							0, 2



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lis	st (No	odes)				Closed
ID	S	G	Н	F	Α	Ρ	State
8	2,0	200	441	641	1,0	1	0,0
							1, 1
							0,1
							1, 2
							1,0
							0, 2

		N8 G 200	
Т	T		
			G

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with $\leq =$ g: continue
9.	else open.add(c)

Ope	Open List (Nodes)						Closed
ID	S	G	Н	F	Α	Ρ	State
8	2,0	200	441	641	1,0	1	0,0
							1, 1
							0,1
							1, 2
							1,0
							0, 2

	-	N8 G 200	
T	Ť		
			G

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	Open List (Nodes)							
ID	S	G	Н	F	Α	Ρ	State	
							0,0	
							1, 1	
							0,1	
							1, 2	
							1,0	
							0, 2	
							2,0	



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lis	st (No	odes)				Closed
ID	S	G	Н	F	Α	Ρ	State
							0,0
							1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0

	-	N8 G 200	<b>C</b> G 300
T	Ť		
			G

L.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
ŧ.	if (node.state in closed) continue
5.	add node to closed
5.	for c in expand(node)
7.	if c.state in closed: continue
3.	if c in open with $\leq =$ g: continue
).	else open.add(c)

Open List (Nodes)						Closed	
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
							1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0

		N8 G 200	
			<b>C</b> G 341
т	Т		
			G

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
5.	for c in expand(node)
7.	if c.state in closed: continue
3.	if c in open with <= g: continue
Э.	else open.add(c)

Open List (Nodes)						Closed	
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
							1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Open List (Nodes)						Closed	
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
10	3,1	341	300	641	1,1	8	1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)



ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
10	3,1	341	300	641	1,1	8	1, 1
							0,1
							1, 2
							1,0
							0, 2
							2, 0



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Open List (Nodes)						Closed	
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
10	3,1	341	300	641	1,1	8	1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Open List (Nodes)							Closed
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
							1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0
							3,1



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with $\leq =$ g: continue
9.	else open.add(c)

Open List (Nodes)						Closed	
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
							1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0
							3, 1



ι.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
1.	if (node.state in closed) continue
5.	add node to closed
5.	for c in expand(node)
7.	if c.state in closed: continue
3.	if c in open with <= g: continue
Э.	else open.add(c)

Ope	en Lis	st (No	des)				Closed
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
							1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0
							3,1



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Open List (Nodes)						Closed	
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
							1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0
							3,1



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)



ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
11	3,2	441	200	641	0,1	10	1, 1
							0,1
							1, 2
							1,0
							0, 2
							2, 0
							3, 1



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lis	st (No	odes)				Closed
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
11	3,2	441	200	641	0,1	10	1, 1
							0,1
							1, 2
							1,0
							0, 2
							2, 0
							3,1



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lis	st (No	odes)				Closed
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
11	3,2	441	200	641	0,1	10	1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0
							3,1



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
1.	if (node.state in closed) continue
5.	add node to closed
5.	for c in expand(node)
7.	if c.state in closed: continue
3.	if c in open with <= g: continue
Э.	else open.add(c)

Ope	en Lis	st (No	des)				Closed
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
							1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0
							3, 1
							3, 2



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lis	st (No	odes)				Closed
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
							1, 1
							0,1
							1, 2
							1,0
							0, 2
							2, 0
							3, 1
							3, 2



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lie	st (No	des)				Closed
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
12	3,3	541	100	641	0,1	11	1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0
							3,1
							3, 2



Open L	. <mark>ist (</mark> N	lodes)
--------	------------------------	--------



ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
12	3,3	541	100	641	0,1	11	1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0
							3, 1
							3, 2

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with $\leq =$ g: continue
9.	else open.add(c)



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

Ope	en Lis	st (No	odes)				Closed
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
12	3,3	541	100	641	0,1	11	1, 1
							0,1
							1, 2
							1,0
							0, 2
							2, 0
							3, 1
							3, 2
		-					
---	---	---	--------------				
Î	t		Î				
			N12 G 541				
			G				

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
1.	if (node.state in closed) continue
5.	add node to closed
5.	for c in expand(node)
7.	if c.state in closed: continue
3.	if c in open with $\leq =$ g: continue
Э.	else open.add(c)

Ope	Closed						
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
							1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0
							3, 1
							3, 2
							3.3

	- <	-	
t	Î		Î
			• N12 G 541
			<b>C</b> G 641

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
1.	if (node.state in closed) continue
5.	add node to closed
5.	for c in expand(node)
7.	if c.state in closed: continue
3.	if c in open with <= g: continue
Э.	else open.add(c)

Ope	Open List (Nodes)						Closed
ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
							1, 1
							0,1
							1, 2
							1,0
							0, 2
							2,0
							3, 1
							3, 2
							3,3



L.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
ŧ.	if (node.state in closed) continue
5.	add node to closed
5.	for c in expand(node)
7.	if c.state in closed: continue
3.	if c in open with $\leq =$ g: continue
Э.	else open.add(c)

С	lose	d

ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
13	3,4	641	0	641	0,1	12	1, 1
							0,1
							1, 2
							1,0
							0,2
							2,0
							3,1
							3, 2
							3,3



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with $\leq =$ g: continue
9.	else open.add(c)

Open	List (	Nod	les)
------	--------	-----	------



ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
13	3,4	641	0	641	0,1	12	1, 1
							0,1
							1, 2
							1,0
							0,2
							2,0
							3,1
							3, 2
							3,3



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)



	_						
ID	S	G	н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
13	3,4	641	0	641	0,1	12	1, 1
14	2,4	682	100	682	-1,1	12	0,1
							1, 2
							1,0
							0, 2
							2,0
							3, 1
							3, 2
							3,3



Open	List	(Nod	les)
------	------	------	------



ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
13	3,4	641	0	641	0,1	12	1, 1
14	2,4	682	100	682	-1,1	12	0,1
							1, 2
							1,0
							0,2
							2,0
							3,1
							3, 2
							3,3

1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)



1.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
4.	if (node.state in closed) continue
5.	add node to closed
6.	for c in expand(node)
7.	if c.state in closed: continue
8.	if c in open with <= g: continue
9.	else open.add(c)

	losea	

ID	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
13	3,4	641	0	641	0,1	12	1, 1
14	2,4	682	100	682	-1,1	12	0,1
							1, 2
							1,0
							0, 2
							2, 0
							3, 1
							3, 2
							3,3



L.	while (!open.empty)
2.	node = remove min f from open
3.	if (node.state is goal) return path
1.	if (node.state in closed) continue
5.	add node to closed
5.	for c in expand(node)
7.	if c.state in closed: continue
3.	if c in open with <= g: continue
€.	else open.add(c)

Ope	en Lis	st (Nc	des)				Closed
D	S	G	Н	F	Α	Ρ	State
9	3,0	300	400	700	1,0	8	0,0
.4	2,4	682	100	682	-1,1	12	1, 1
							0,1
							1, 2
		Goal	Ехра	andec	!!		1,0
	R	0,2					
		2,0					
		3,1					
							3, 2

3,3