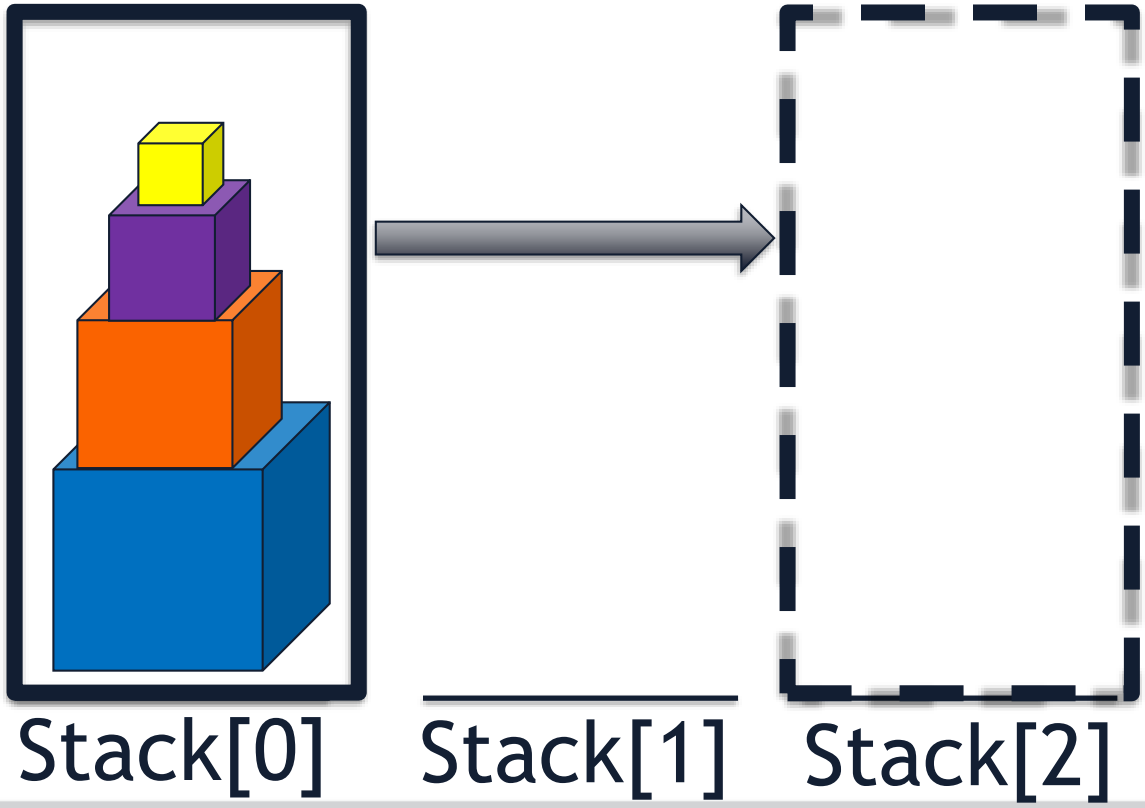
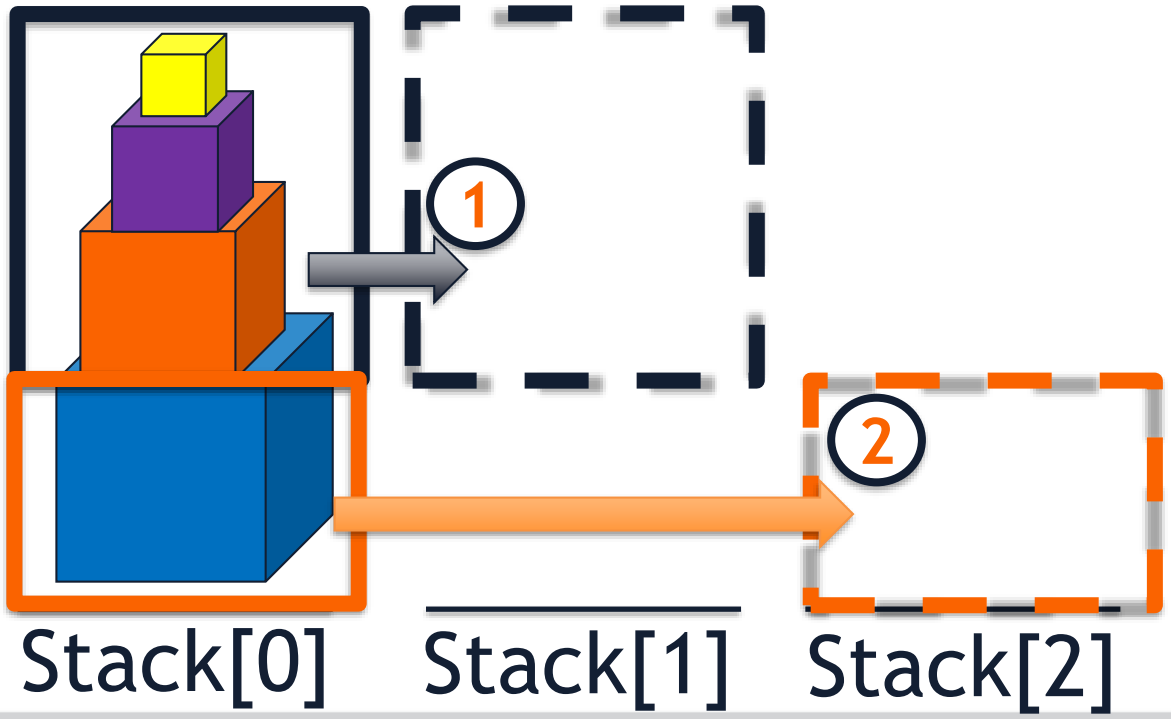
Tower of Hanoi – Solution #2

Prof. Wade Fagen-Ulmschneider

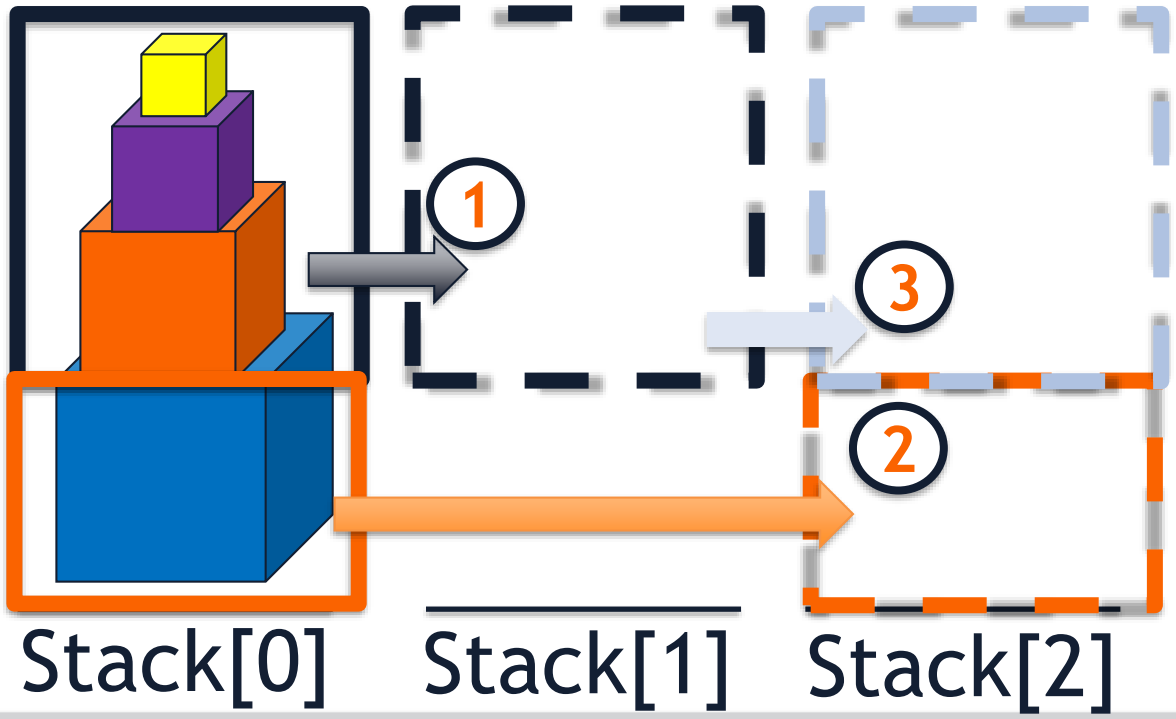
Moves Made:



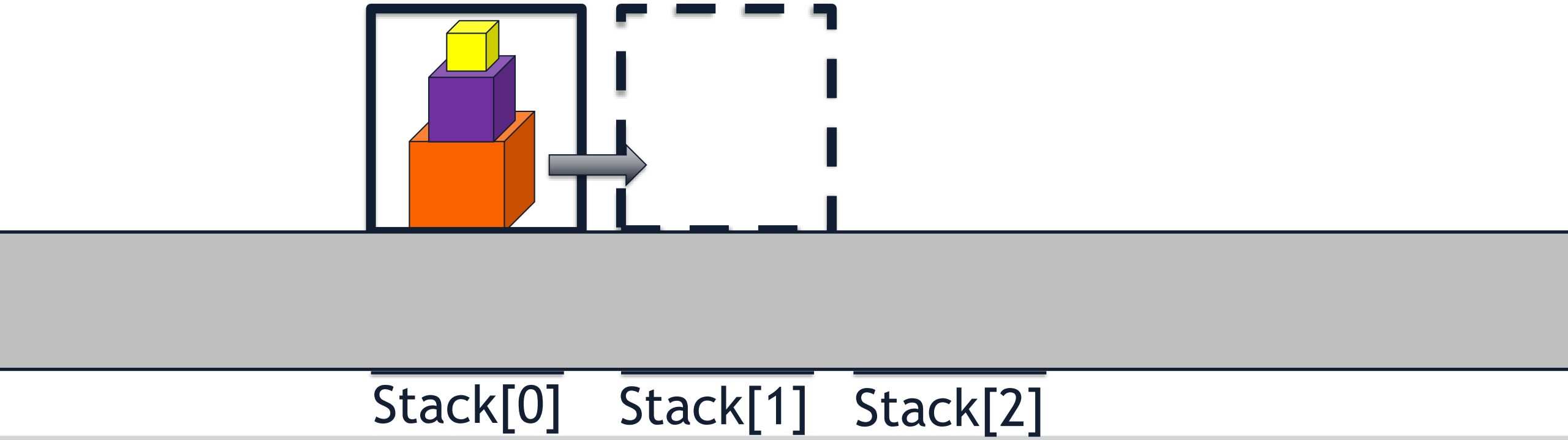
Moves Made:



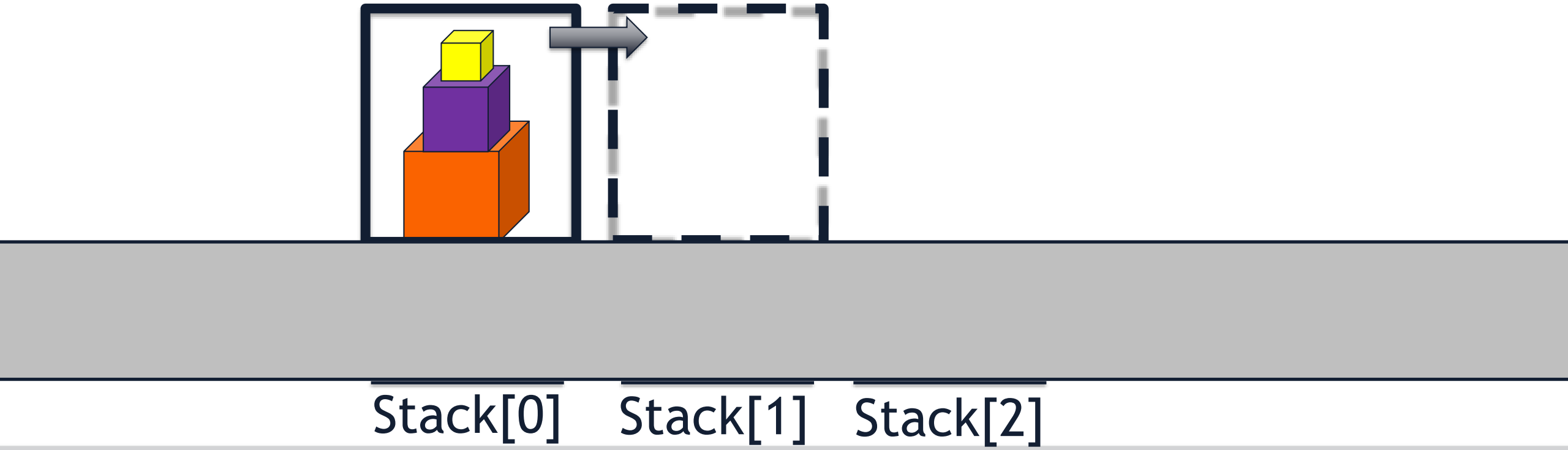
Moves Made:



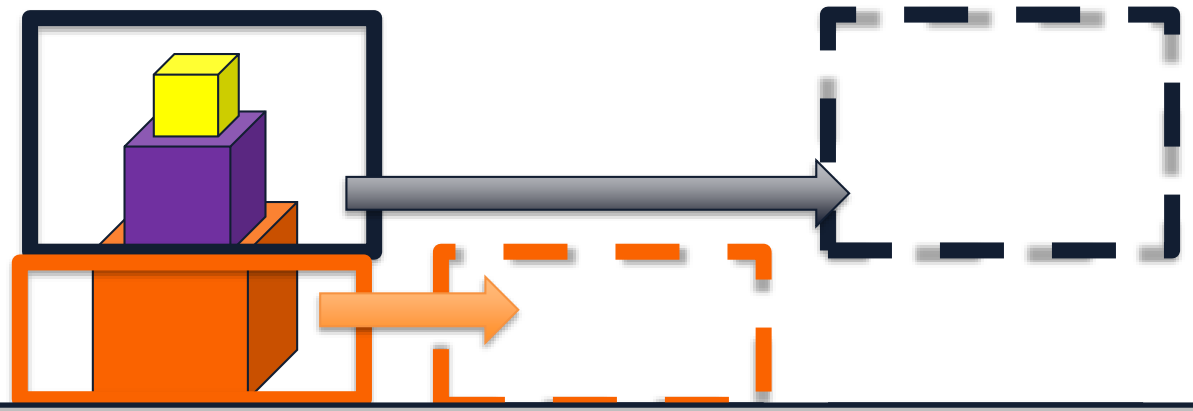
Moves Made:



Moves Made:

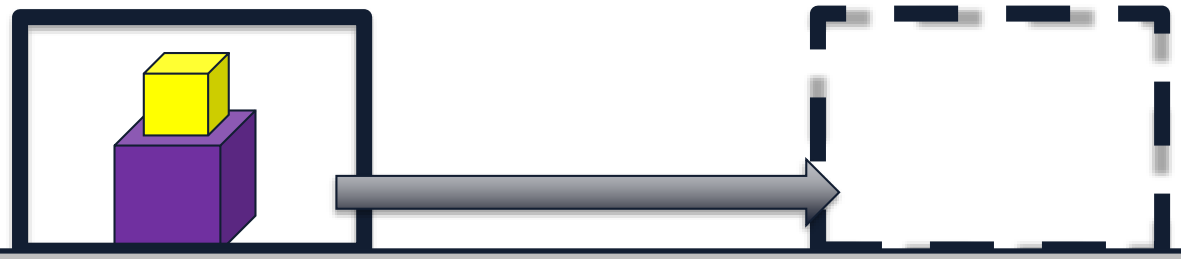


Moves Made:



Stack[0] Stack[1] Stack[2]

Moves Made:

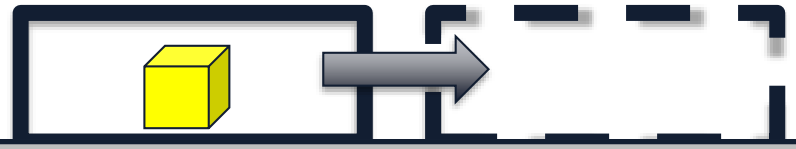


Stack[0] Stack[1] Stack[2]

Moves Made:

Stack[0] Stack[1] Stack[2]

Moves Made:



Stack[0] Stack[1] Stack[2]

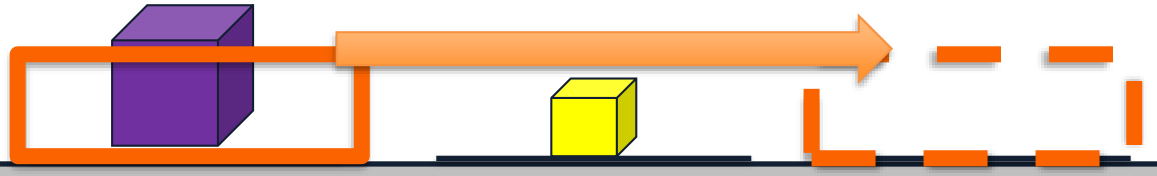
Moves Made:

0 ➔ 1

Stack[0] Stack[1] Stack[2]

Moves Made:

0 ➔ 1

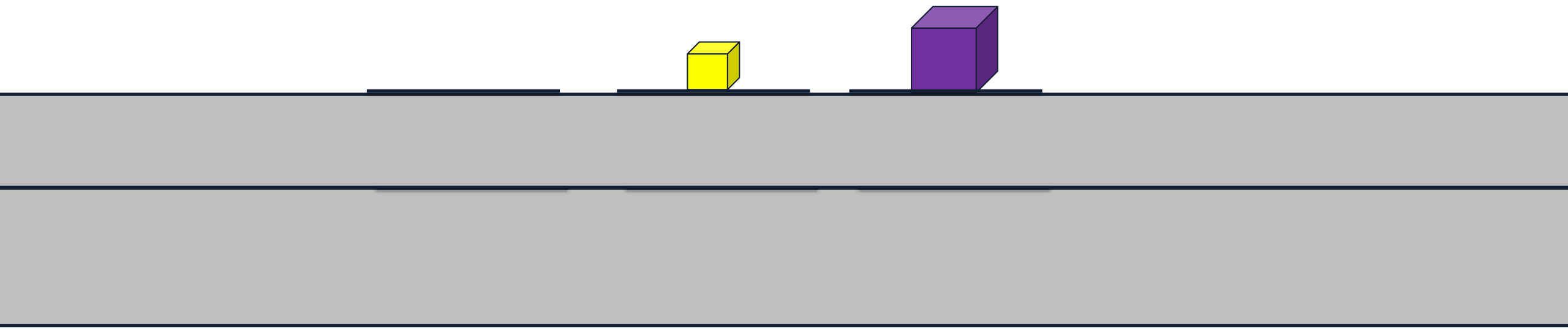


Stack[0] Stack[1] Stack[2]

Moves Made:

0 → 1

0 → 2

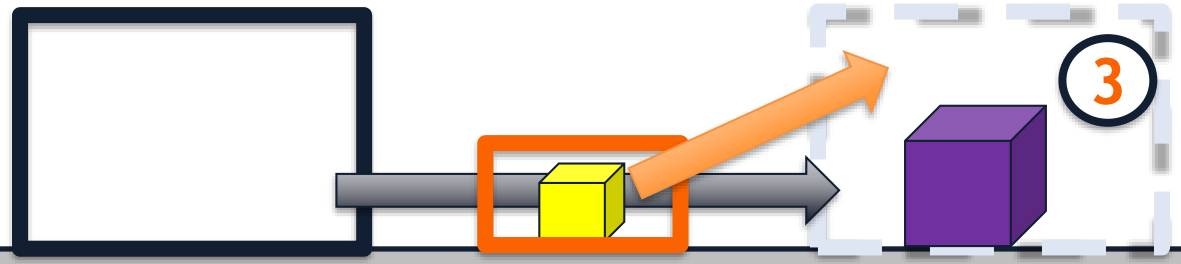


Stack[0] Stack[1] Stack[2]

Moves Made:

0 → 1

0 → 2



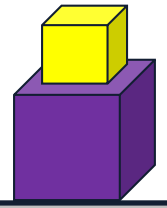
Stack[0] Stack[1] Stack[2]

Moves Made:

0 → 1

0 → 2

1 → 2



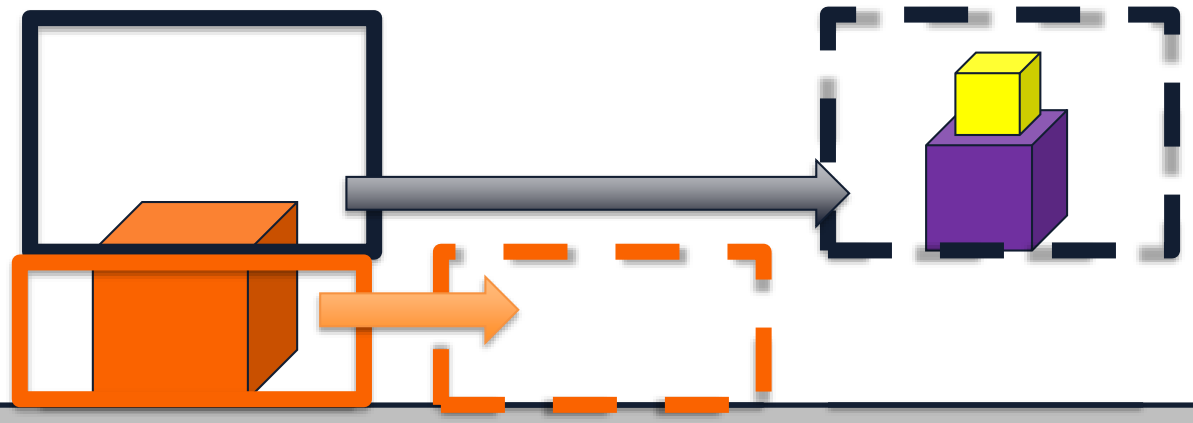
Stack[0] Stack[1] Stack[2]

Moves Made:

0 → 1

0 → 2

1 → 2



Stack[0] Stack[1] Stack[2]

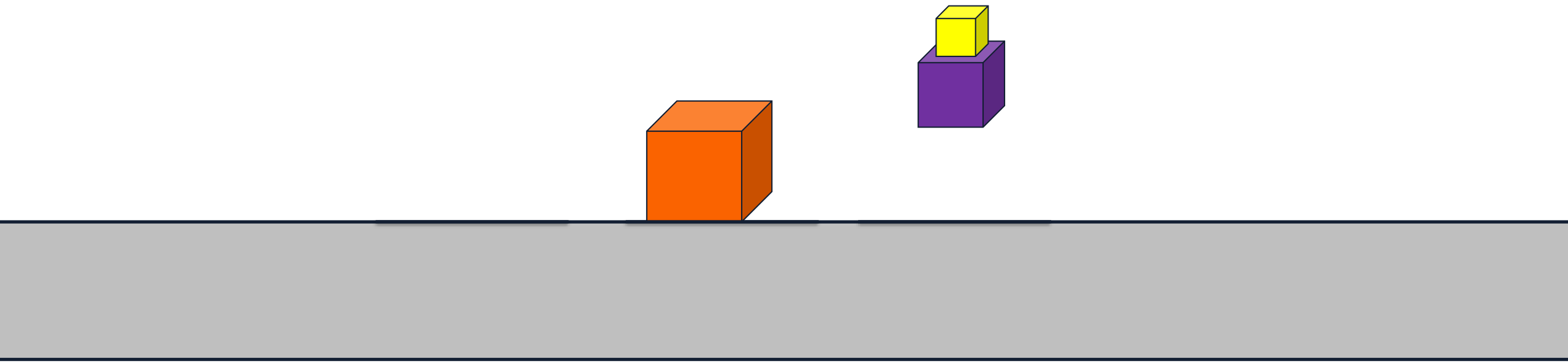
Moves Made:

0 → 1

0 → 2

1 → 2

0 → 1



Stack[0] Stack[1] Stack[2]

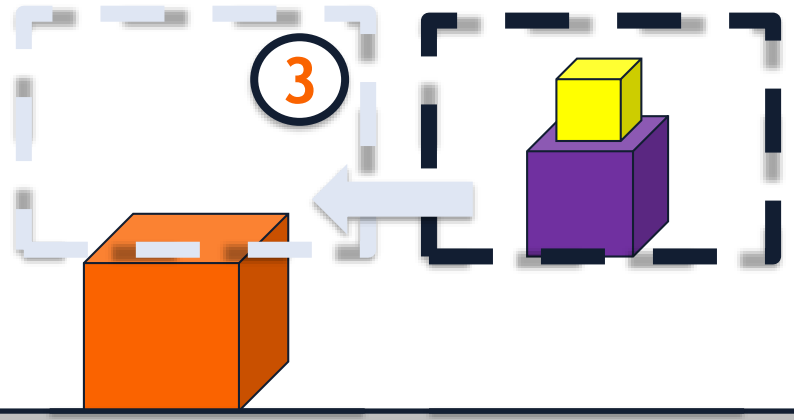
Moves Made:

0 → 1

0 → 2

1 → 2

0 → 1



Stack[0] Stack[1] Stack[2]

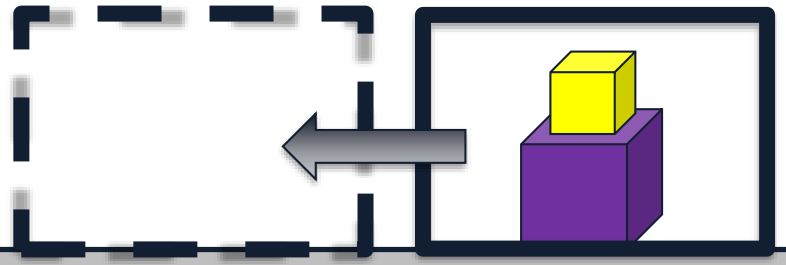
Moves Made:

0 → 1

0 → 2

1 → 2

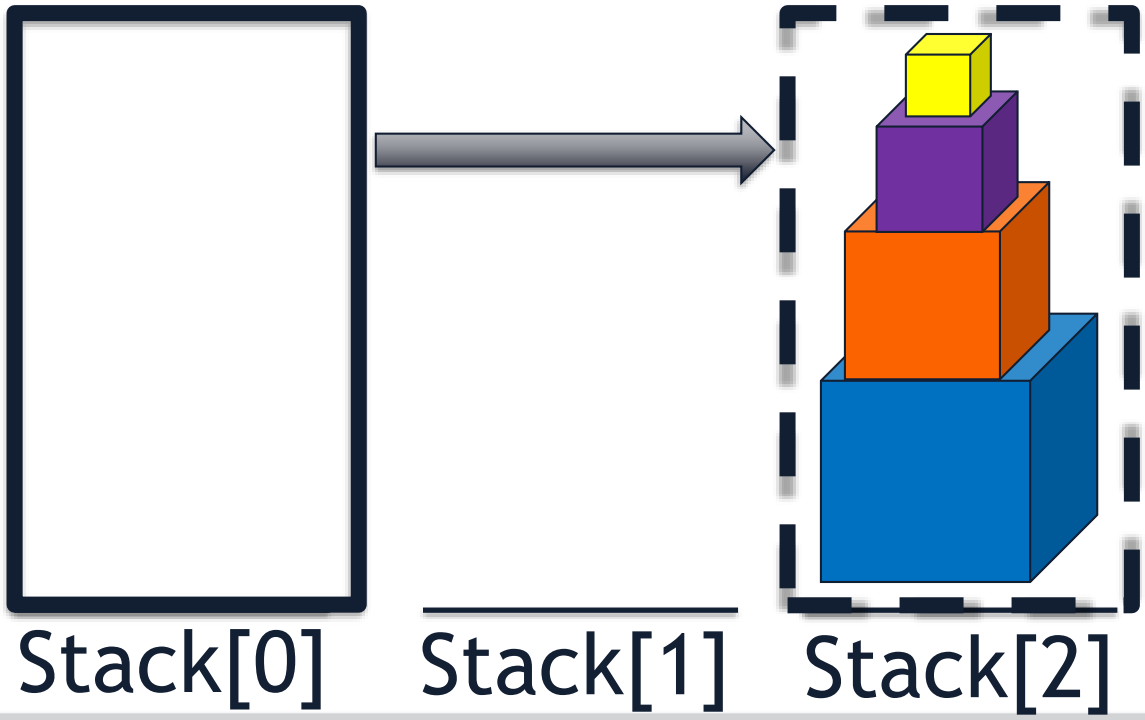
0 → 1



Stack[0] Stack[1] Stack[2]

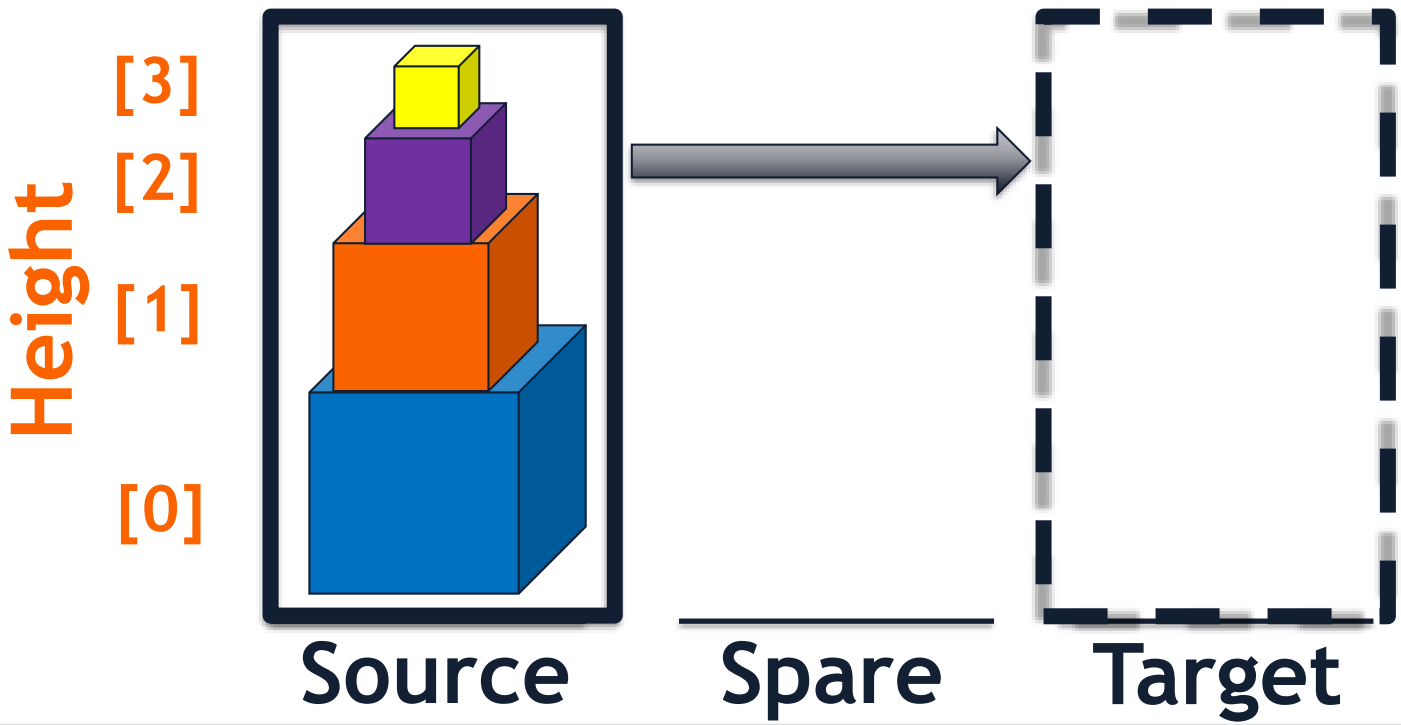
…unrolling, and recusing, and unrolling…

…many moves later…


Stack[0]
Stack[1]
Stack[2]

Moves Planned:

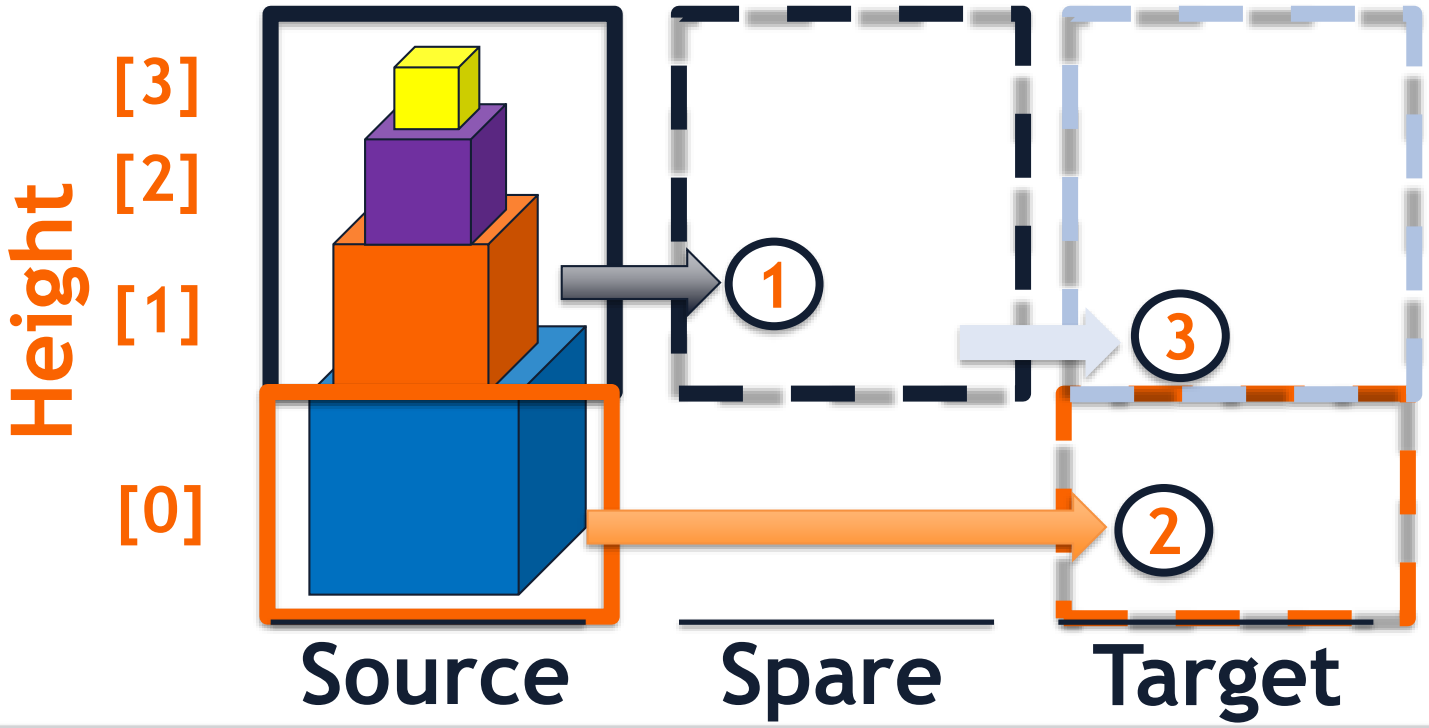
move(Source[0…3] → Target)



Moves Planned:

move(Source[0…3] → Target)

- move(Source[1…3] → Spare)
- move(Source[0] → Target)
- move(Spare[1…3] → Target)



Moves Planned:

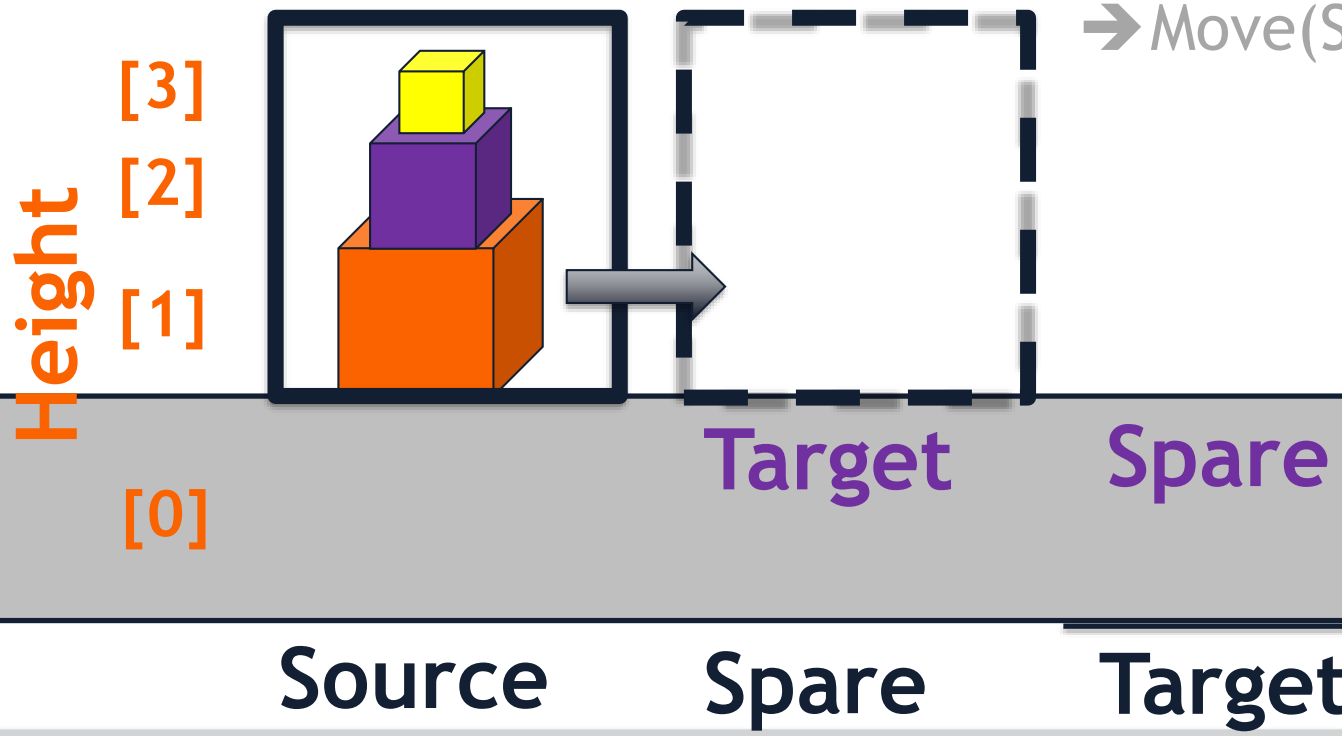
move(Source[0…3] → Target)

➔ move(Source[1…3] → Spare)

SWAP: *Old "Spare" is now "Target"!*

➔ move(Source[0] → Target)

➔ Move(Spare[1…3] → Target)



Moves Planned:

move(Source[0…3] → Target)

→ move(Source[1…3] → Spare)

SWAP: Old "Spare" is now "Target"!

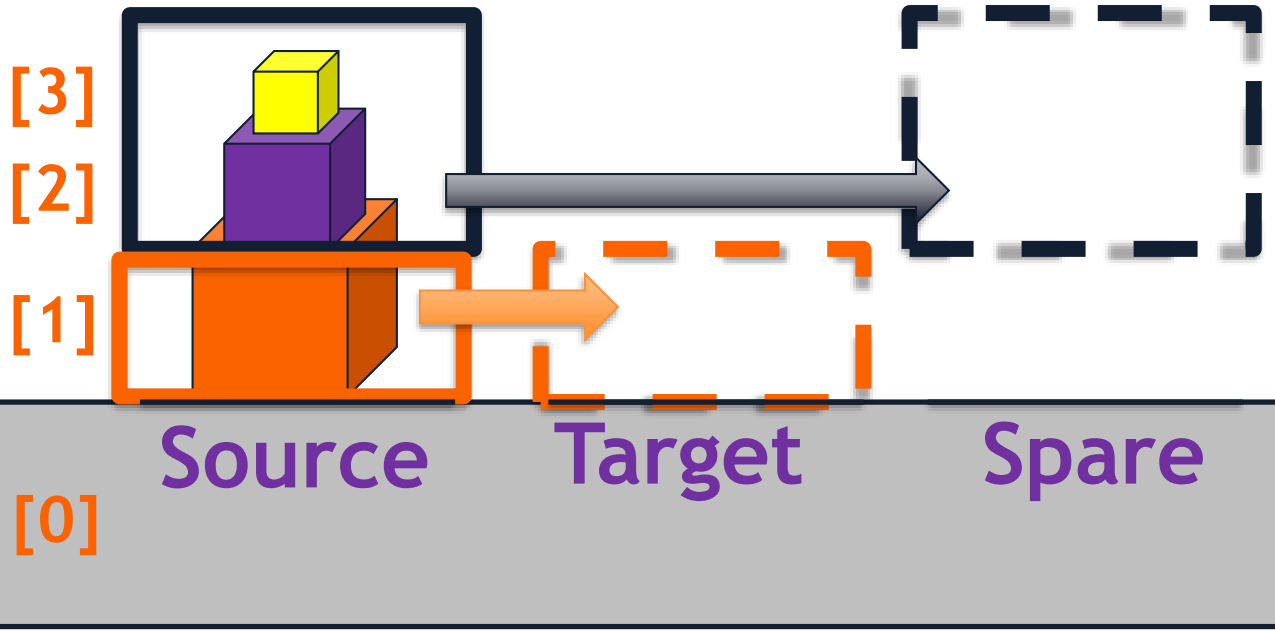
→ move(Source[2…3] → Spare)

→ move(Source[1] → Target)

→ Move(Spare[1…3] → Target)

→ move(Source[0] → Target)

→ Move(Spare[1…3] → Target)



Moves Planned:

move(Source[0…3] → Target)

→ move(Source[1…3] → Spare)

→ move(Source[2…3] → Spare)

→ move(Source[3…3] → Spare)

→ move(Source[2] → Target)

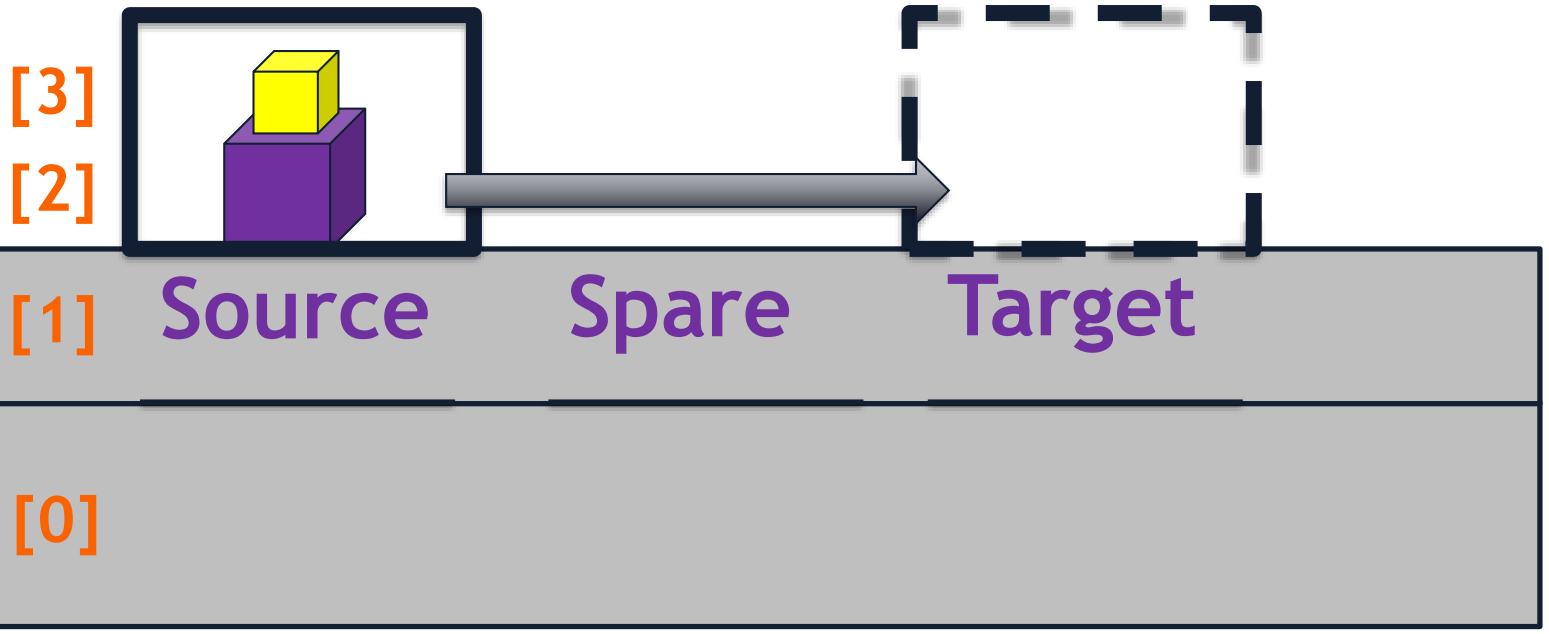
→ Move(Spare[3…3] → Target)

→ move(Source[1] → Target)

→ Move(Spare[1…3] → Target)

→ move(Source[0] → Target)

→ Move(Spare[1…3] → Target)



Moves Planned:

move(Source[0…3] → Target)

- ➔move(Source[1…3] → Spare)
  - ➔move(Source[2…3] → Spare)
    - ➔move(Source[3…3] → Spare) ➔ ...
    - ➔move(Source[2] → Target) ➔ ...
    - ➔Move(Spare[3…3] → Target) ➔ ...
  - ➔move(Source[1] → Target) ➔ ...
  - ➔Move(Spare[1…3] → Target) ➔ ...
- ➔move(Source[0] → Target) ➔ ...
- ➔Move(Spare[1…3] → Target) ➔ ...

Moves Planned:

move(Source[0…3] → Target)

➔move(Source[1…3] → Spare) ➔ ...

➔move(Source[0] → Target) ➔ ...

➔Move(Spare[1…3] → Target) ➔ ...

Moves Planned:

move(Source[start…end] → Target)

➔move(Source[(start+1)…end] → Spare) ➔ ...

➔move(Source[start] → Target) ➔ ...

➔Move(Spare[(start+1)…end] → Target) ➔ ...

cpp-tower-solution2/Game.cpp

```
45 void Game::_move(
46   unsigned start, unsigned end,
47   Stack & source, Stack & target, Stack & spare,
48   unsigned depth
49 ) {
50   cout << "Planning (depth=" << depth++ << "): Move [" << /* ... */
51
52   // Check if we are only moving one cube:
53   if (start == end) {
54     // If so, move it directly:
55     _moveCube( source, target );
56     cout << *this << endl;
57   } else {
58     // Otherwise, use our move strategy:
59     _move(start + 1, end   , source, spare , target, depth);
60     _move(start    , start, source, target, spare , depth);
61     _move(start + 1, end   , spare , target, source, depth);
62   }
63 }
```