

1 (move d1 d2 pole3)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 d2 pole3)
:precondition
  (and
    (smaller d1 pole3)
    (on d1 d2)
    (clear d1)
    (clear pole3)
  )
:effect
  (and
    (clear d2)
    (on d1 pole3)
    (not
      (on d1 d2)
    )
    (not
      (clear pole3)
    )
  )
)
```

2 (move d2 d3 pole2)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d2 d3 pole2)
:precondition
  (and
    (smaller d2 pole2)
    (on d2 d3)
    (clear d2)
    (clear pole2)
  )
:effect
  (and
    (clear d3)
    (on d2 pole2)
    (not
      (on d2 d3)
    )
    (not
      (clear pole2)
    )
  )
)
```

3 (move d1 pole3 d2)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 pole3 d2)
:precondition
  (and
    (smaller d1 d2)
    (on d1 pole3)
    (clear d1)
    (clear d2)
  )
:effect
  (and
    (clear pole3)
    (on d1 d2)
    (not
      (on d1 pole3)
    )
    (not
      (clear d2)
    )
  )
)
```

4 (move d3 d4 pole3)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d3 d4 pole3)
:precondition
  (and
    (smaller d3 pole3)
    (on d3 d4)
    (clear d3)
    (clear pole3)
  )
:effect
  (and
    (clear d4)
    (on d3 pole3)
    (not
      (on d3 d4)
    )
    (not
      (clear pole3)
    )
  )
)
```

5 (move d1 d2 d4)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 d2 d4)
:precondition
  (and
    (smaller d1 d4)
    (on d1 d2)
    (clear d1)
    (clear d4)
  )
:effect
  (and
    (clear d2)
    (on d1 d4)
    (not
      (on d1 d2)
    )
    (not
      (clear d4)
    )
  )
)
```

6 (move d2 pole2 d3)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d2 pole2 d3)
:precondition
  (and
    (smaller d2 d3)
    (on d2 pole2)
    (clear d2)
    (clear d3)
  )
:effect
  (and
    (clear pole2)
    (on d2 d3)
    (not
      (on d2 pole2)
    )
    (not
      (clear d3)
    )
  )
)
```

7 (move d1 d4 d2)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 d4 d2)
:precondition
  (and
    (smaller d1 d2)
    (on d1 d4)
    (clear d1)
    (clear d2)
  )
:effect
  (and
    (clear d4)
    (on d1 d2)
    (not
      (on d1 d4)
    )
    (not
      (clear d2)
    )
  )
)
```

8 (move d4 d5 pole2)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d4 d5 pole2)
:precondition
  (and
    (smaller d4 pole2)
    (on d4 d5)
    (clear d4)
    (clear pole2)
  )
:effect
  (and
    (clear d5)
    (on d4 pole2)
    (not
      (on d4 d5)
    )
    (not
      (clear pole2)
    )
  )
)
```


9 (move d1 d2 d4)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 d2 d4)
:precondition
  (and
    (smaller d1 d4)
    (on d1 d2)
    (clear d1)
    (clear d4)
  )
:effect
  (and
    (clear d2)
    (on d1 d4)
    (not
      (on d1 d2)
    )
    (not
      (clear d4)
    )
  )
)
```

10 (move d2 d3 d5)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d2 d3 d5)
:precondition
  (and
    (smaller d2 d5)
    (on d2 d3)
    (clear d2)
    (clear d5)
  )
:effect
  (and
    (clear d3)
    (on d2 d5)
    (not
      (on d2 d3)
    )
    (not
      (clear d5)
    )
  )
)
```

11 (move d1 d4 d2)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 d4 d2)
:precondition
  (and
    (smaller d1 d2)
    (on d1 d4)
    (clear d1)
    (clear d2)
  )
:effect
  (and
    (clear d4)
    (on d1 d2)
    (not
      (on d1 d4)
    )
    (not
      (clear d2)
    )
  )
)
```

12 (move d3 pole3 d4)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d3 pole3 d4)
:precondition
  (and
    (smaller d3 d4)
    (on d3 pole3)
    (clear d3)
    (clear d4)
  )
:effect
  (and
    (clear pole3)
    (on d3 d4)
    (not
      (on d3 pole3)
    )
    (not
      (clear d4)
    )
  )
)
```

13 (move d1 d2 d3)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 d2 d3)
:precondition
  (and
    (smaller d1 d3)
    (on d1 d2)
    (clear d1)
    (clear d3)
  )
:effect
  (and
    (clear d2)
    (on d1 d3)
    (not
      (on d1 d2)
    )
    (not
      (clear d3)
    )
  )
)
```

14 (move d2 d5 pole3)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d2 d5 pole3)
:precondition
  (and
    (smaller d2 pole3)
    (on d2 d5)
    (clear d2)
    (clear pole3)
  )
:effect
  (and
    (clear d5)
    (on d2 pole3)
    (not
      (on d2 d5)
    )
    (not
      (clear pole3)
    )
  )
)
```

15 (move d1 d3 d5)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 d3 d5)
:precondition
  (and
    (smaller d1 d5)
    (on d1 d3)
    (clear d1)
    (clear d5)
  )
:effect
  (and
    (clear d3)
    (on d1 d5)
    (not
      (on d1 d3)
    )
    (not
      (clear d5)
    )
  )
)
```

16 (move d2 pole3 d3)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d2 pole3 d3)
:precondition
  (and
    (smaller d2 d3)
    (on d2 pole3)
    (clear d2)
    (clear d3)
  )
:effect
  (and
    (clear pole3)
    (on d2 d3)
    (not
      (on d2 pole3)
    )
    (not
      (clear d3)
    )
  )
)
```


17 (move d1 d5 d2)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 d5 d2)
:precondition
  (and
    (smaller d1 d2)
    (on d1 d5)
    (clear d1)
    (clear d2)
  )
:effect
  (and
    (clear d5)
    (on d1 d2)
    (not
      (on d1 d5)
    )
    (not
      (clear d2)
    )
  )
)
```

18 (move d5 pole1 pole3)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d5 pole1 pole3)
:precondition
  (and
    (smaller d5 pole3)
    (on d5 pole1)
    (clear d5)
    (clear pole3)
  )
:effect
  (and
    (clear pole1)
    (on d5 pole3)
    (not
      (on d5 pole1)
    )
    (not
      (clear pole3)
    )
  )
)
```

19 (move d1 d2 pole1)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 d2 pole1)
:precondition
  (and
    (smaller d1 pole1)
    (on d1 d2)
    (clear d1)
    (clear pole1)
  )
:effect
  (and
    (clear d2)
    (on d1 pole1)
    (not
      (on d1 d2)
    )
    (not
      (clear pole1)
    )
  )
)
```

20 (move d2 d3 d5)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d2 d3 d5)
:precondition
  (and
    (smaller d2 d5)
    (on d2 d3)
    (clear d2)
    (clear d5)
  )
:effect
  (and
    (clear d3)
    (on d2 d5)
    (not
      (on d2 d3)
    )
    (not
      (clear d5)
    )
  )
)
```

21 (move d1 pole1 d2)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 pole1 d2)
:precondition
  (and
    (smaller d1 d2)
    (on d1 pole1)
    (clear d1)
    (clear d2)
  )
:effect
  (and
    (clear pole1)
    (on d1 d2)
    (not
      (on d1 pole1)
    )
    (not
      (clear d2)
    )
  )
)
```

22 (move d3 d4 pole1)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d3 d4 pole1)
:precondition
  (and
    (smaller d3 pole1)
    (on d3 d4)
    (clear d3)
    (clear pole1)
  )
:effect
  (and
    (clear d4)
    (on d3 pole1)
    (not
      (on d3 d4)
    )
    (not
      (clear pole1)
    )
  )
)
```

23 (move d1 d2 d4)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 d2 d4)
:precondition
  (and
    (smaller d1 d4)
    (on d1 d2)
    (clear d1)
    (clear d4)
  )
:effect
  (and
    (clear d2)
    (on d1 d4)
    (not
      (on d1 d2)
    )
    (not
      (clear d4)
    )
  )
)
```

24 (move d2 d5 d3)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d2 d5 d3)
:precondition
  (and
    (smaller d2 d3)
    (on d2 d5)
    (clear d2)
    (clear d3)
  )
:effect
  (and
    (clear d5)
    (on d2 d3)
    (not
      (on d2 d5)
    )
    (not
      (clear d3)
    )
  )
)
```


25 (move d1 d4 d2)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 d4 d2)
:precondition
  (and
    (smaller d1 d2)
    (on d1 d4)
    (clear d1)
    (clear d2)
  )
:effect
  (and
    (clear d4)
    (on d1 d2)
    (not
      (on d1 d4)
    )
    (not
      (clear d2)
    )
  )
)
```

26 (move d4 pole2 d5)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d4 pole2 d5)
:precondition
  (and
    (smaller d4 d5)
    (on d4 pole2)
    (clear d4)
    (clear d5)
  )
:effect
  (and
    (clear pole2)
    (on d4 d5)
    (not
      (on d4 pole2)
    )
    (not
      (clear d5)
    )
  )
)
```

27 (move d1 d2 d4)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 d2 d4)
:precondition
  (and
    (smaller d1 d4)
    (on d1 d2)
    (clear d1)
    (clear d4)
  )
:effect
  (and
    (clear d2)
    (on d1 d4)
    (not
      (on d1 d2)
    )
    (not
      (clear d4)
    )
  )
)
```

28 (move d2 d3 pole2)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d2 d3 pole2)
:precondition
  (and
    (smaller d2 pole2)
    (on d2 d3)
    (clear d2)
    (clear pole2)
  )
:effect
  (and
    (clear d3)
    (on d2 pole2)
    (not
      (on d2 d3)
    )
    (not
      (clear pole2)
    )
  )
)
```

29 (move d1 d4 d2)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 d4 d2)
:precondition
  (and
    (smaller d1 d2)
    (on d1 d4)
    (clear d1)
    (clear d2)
  )
:effect
  (and
    (clear d4)
    (on d1 d2)
    (not
      (on d1 d4)
    )
    (not
      (clear d2)
    )
  )
)
```

30 (move d3 pole1 d4)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d3 pole1 d4)
:precondition
  (and
    (smaller d3 d4)
    (on d3 pole1)
    (clear d3)
    (clear d4)
  )
:effect
  (and
    (clear pole1)
    (on d3 d4)
    (not
      (on d3 pole1)
    )
    (not
      (clear d4)
    )
  )
)
```

31 (move d1 d2 pole1)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d1 d2 pole1)
:precondition
  (and
    (smaller d1 pole1)
    (on d1 d2)
    (clear d1)
    (clear pole1)
  )
:effect
  (and
    (clear d2)
    (on d1 pole1)
    (not
      (on d1 d2)
    )
    (not
      (clear pole1)
    )
  )
)
```

32 (move d2 pole2 d3)

Found Plan (output)

(move d1 d2 pole3)

(move d2 d3 pole2)

(move d1 pole3 d2)

(move d3 d4 pole3)

(move d1 d2 d4)

(move d2 pole2 d3)

(move d1 d4 d2)

(move d4 d5 pole2)

(move d1 d2 d4)

(move d2 d3 d5)

```
(:action move
:parameters (d2 pole2 d3)
:precondition
  (and
    (smaller d2 d3)
    (on d2 pole2)
    (clear d2)
    (clear d3)
  )
:effect
  (and
    (clear pole2)
    (on d2 d3)
    (not
      (on d2 pole2)
    )
    (not
      (clear d3)
    )
  )
)
```


Found Plan (output)

(move d1 d2 pole3)
(move d2 d3 pole2)
(move d1 pole3 d2)
(move d3 d4 pole3)
(move d1 d2 d4)
(move d2 pole2 d3)
(move d1 d4 d2)
(move d4 d5 pole2)
(move d1 d2 d4)
(move d2 d3 d5)

```
(:action move
:parameters (d1 pole1 d2)
:precondition
  (and
    (smaller d1 d2)
    (on d1 pole1)
    (clear d1)
    (clear d2)
  )
:effect
  (and
    (clear pole1)
    (on d1 d2)
    (not
      (on d1 pole1)
    )
    (not
      (clear d2)
    )
  )
)
```