# Snap Plan Test Bench

Dependencies

In [ ]:

```
! pip install ortools
! pip install plotly
```

## Imports

In [1]:

```
from ortools.sat.python import cp_model

from planner.io.parser import parse_floor
from planner.io.visualizer import visualize_floor
from planner.solver import plan_floor

import time
```

## Functions

In [2]:

```
def test_run(input_path: str, max_num_of_solutions: int):
    floor = parse_floor(input_path)
    print('Started search...')

    start_time = time.time()
    status, solutions = plan_floor(floor, max_num_of_solutions)

    if status == cp_model.OPTIMAL:
        print(f'OPTIMAL\nNumber of solutions is: {len(solutions)}')
    elif status == cp_model.FEASIBLE:
        print(f'FEASIBLE\nNumber of solutions is: {len(solutions)}')
    else:
        print('INFEASIBLE.')

    end_time = time.time()
    print(f'Done in {round(end_time - start_time, 2)} seconds')

    return solutions
```

In [3]:

```
def visualize_solution(solutions, solution_number):
    solution = solutions[solution_number - 1]
    print('Soft Constraints:')
    print('-----------------')
    for soft_constraint, value in solution['scores'].items():
        print(f'Soft constraint {soft_constraint}: {value}')
    visualize_floor(solution)
```

Test Runs

Input 1

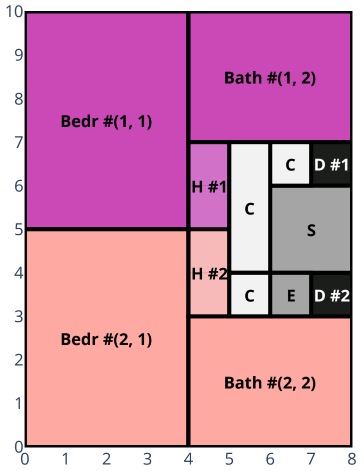
In [4]:

```
solutions = test_run('inputs/floor_input_1.json', 100)
```

```
Started search...
OPTIMAL
Number of solutions is: 100
Done in 19.07 seconds
```

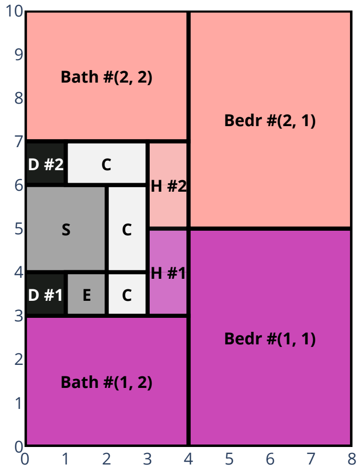
In [5]:
```
visualize_solution(solutions, 1)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



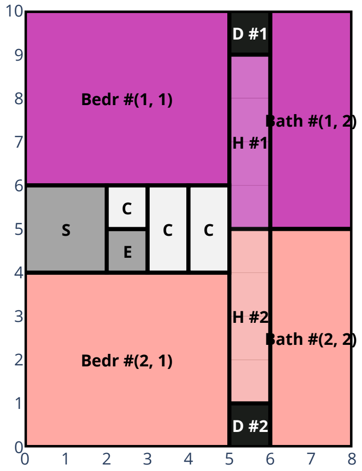
In [6]:
```
visualize_solution(solutions, 20)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



```
In [7]: visualize_solution(solutions, 100)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



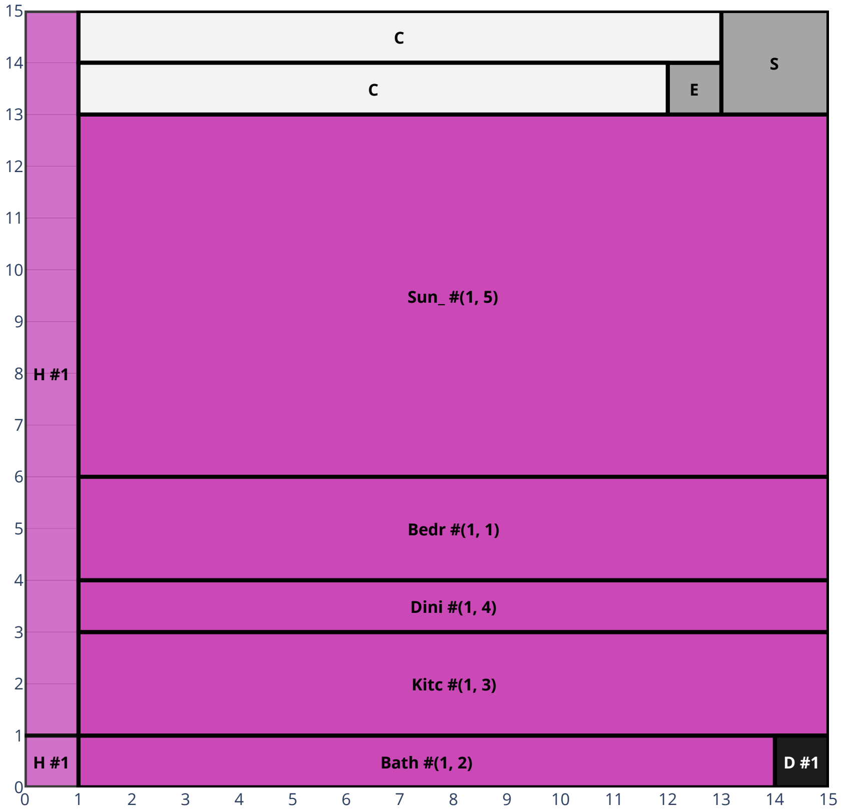
Input 2

```
In [8]: solutions = test_run('inputs/floor_input_2.json', 100)
```

```
Started search...
OPTIMAL
Number of solutions is: 100
Done in 41.31 seconds
```

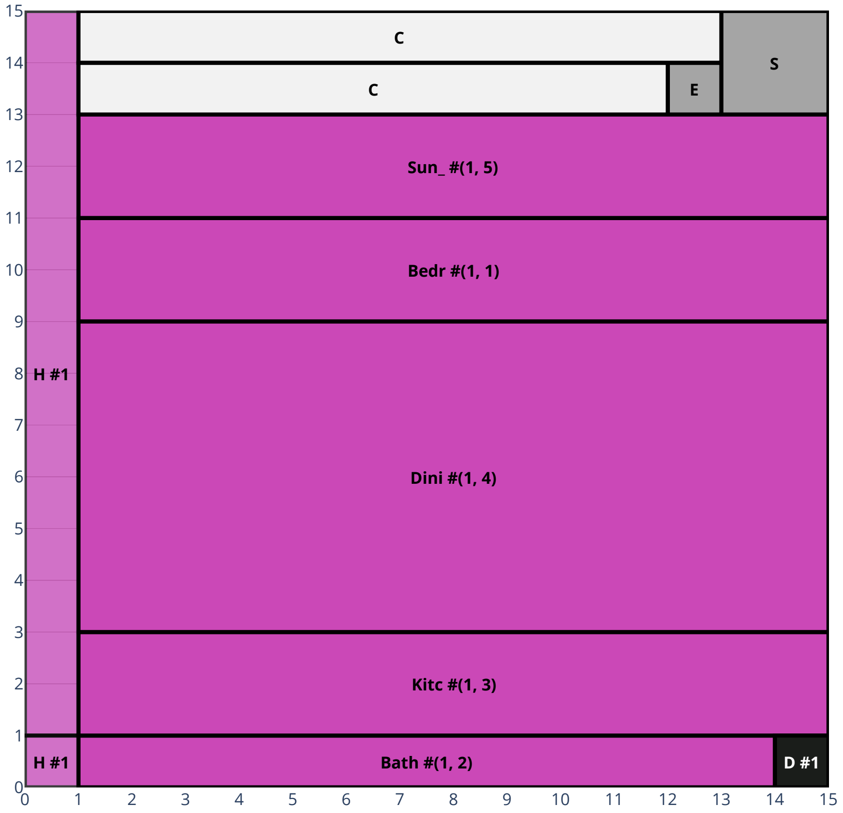
```
In [9]: visualize_solution(solutions, 1)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



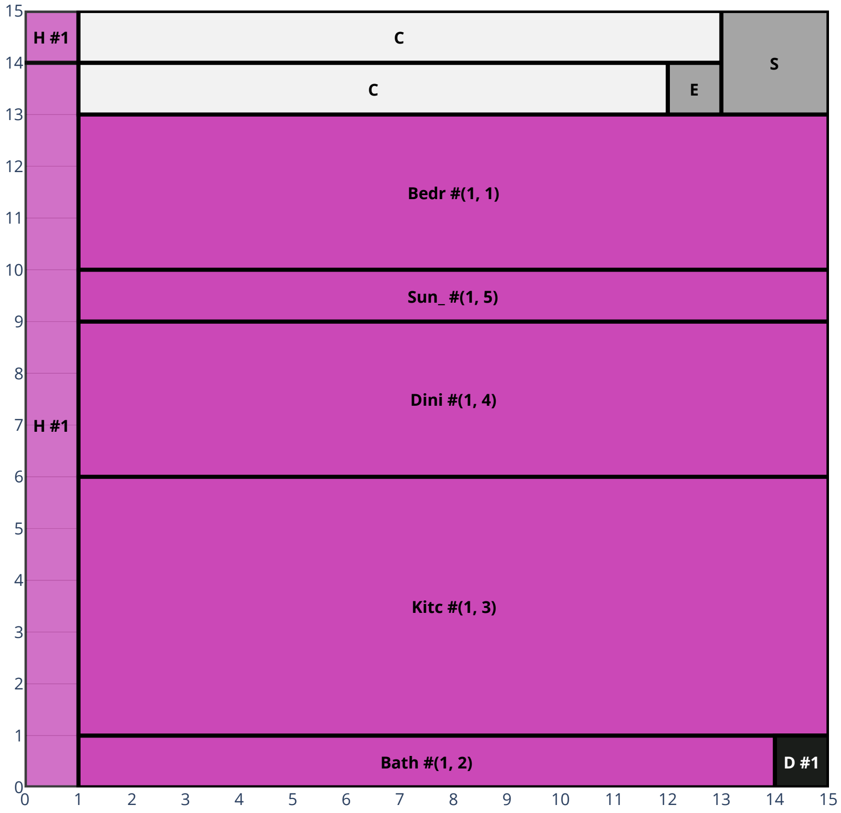
In [10]:
```
visualize_solution(solutions, 50)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



```
In [11]: visualize_solution(solutions, 100)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



Input 3

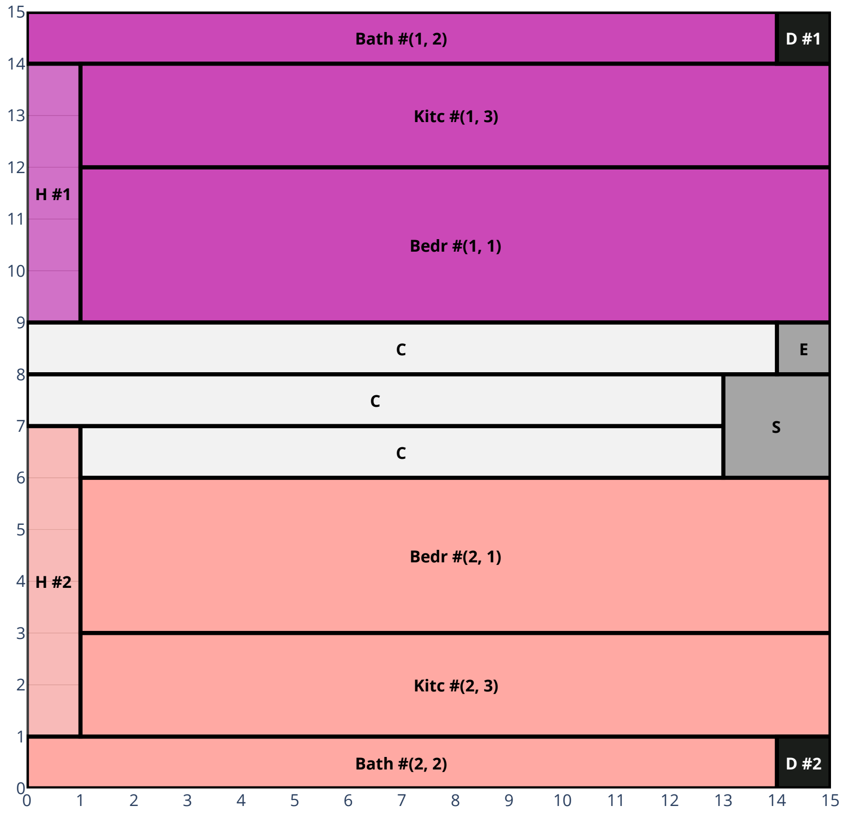
```
In [12]: solutions = test_run('inputs/floor_input_3.json', 100)
```

```
Started search...
OPTIMAL
Number of solutions is: 100
Done in 29.26 seconds
```

In [13]:

```
visualize_solution(solutions, 1)
```

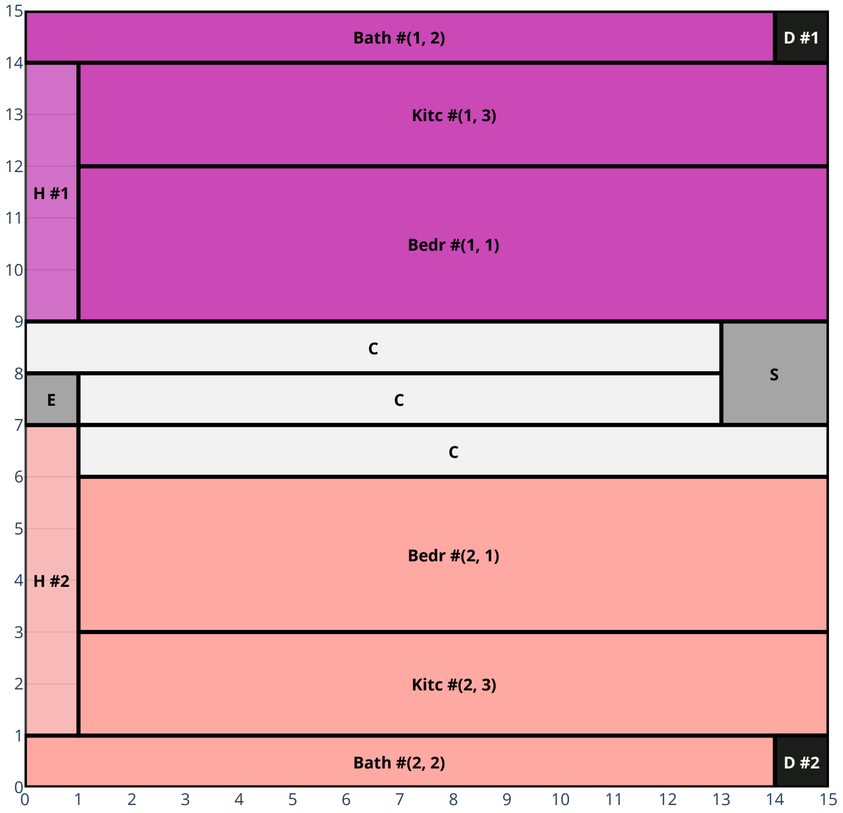
```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



In [14]:

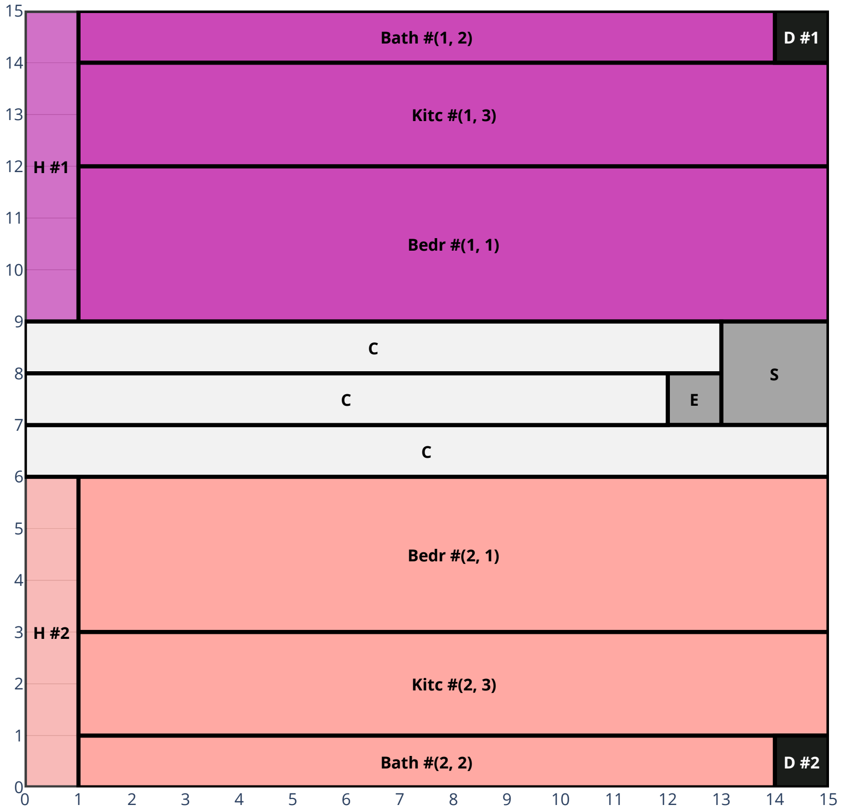
```
visualize_solution(solutions, 50)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



In [15]:
```
visualize_solution(solutions, 100)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



Input 4

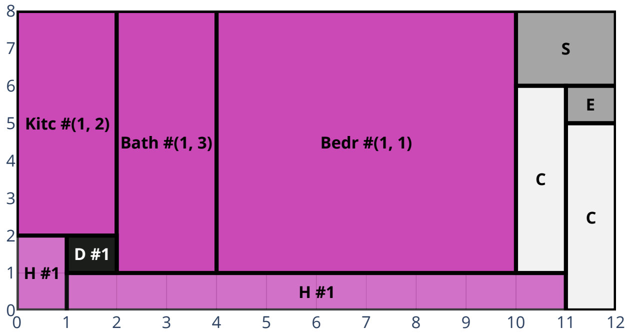
In [16]:
```
solutions = test_run('inputs/floor_input_4.json', 100)
```

```
Started search...
OPTIMAL
Number of solutions is: 100
Done in 15.04 seconds
```

In [17]:

```
visualize_solution(solutions, 1)
```

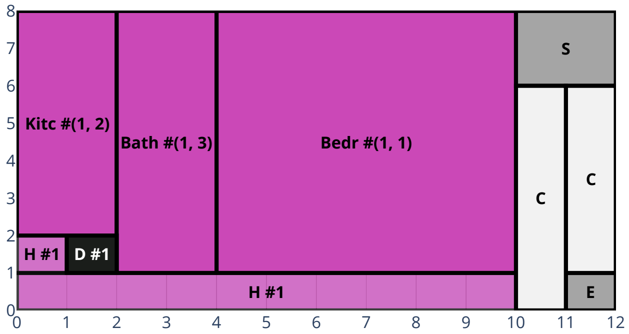
```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



In [18]:

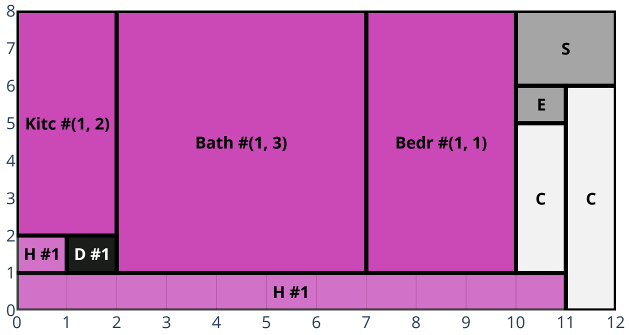
```
visualize_solution(solutions, 50)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



```
In [19]: visualize_solution(solutions, 100)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



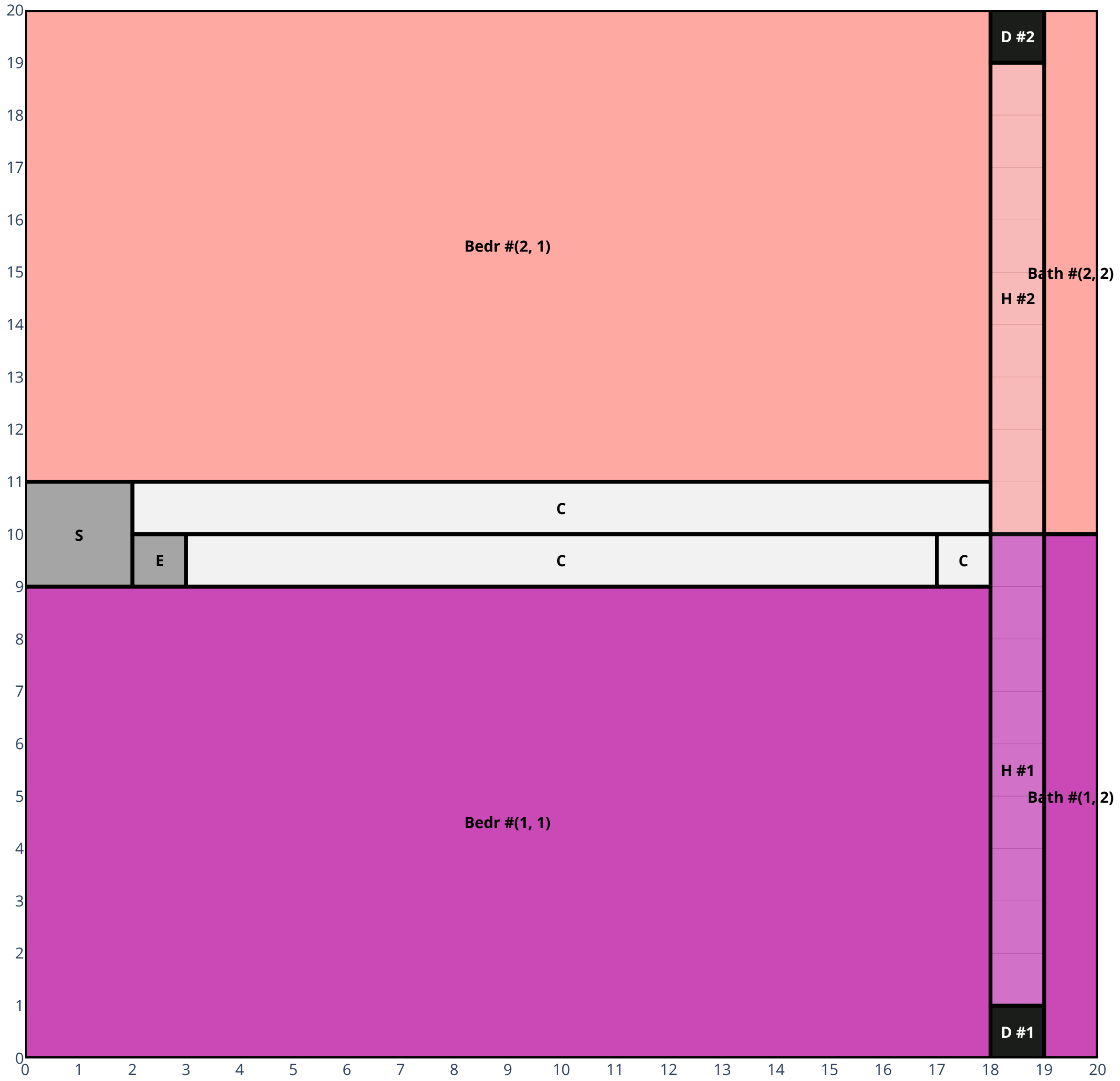
Input 5

```
In [20]: solutions = test_run('inputs/floor_input_5.json', 100)
```

```
Started search...
OPTIMAL
Number of solutions is: 100
Done in 61.3 seconds
```

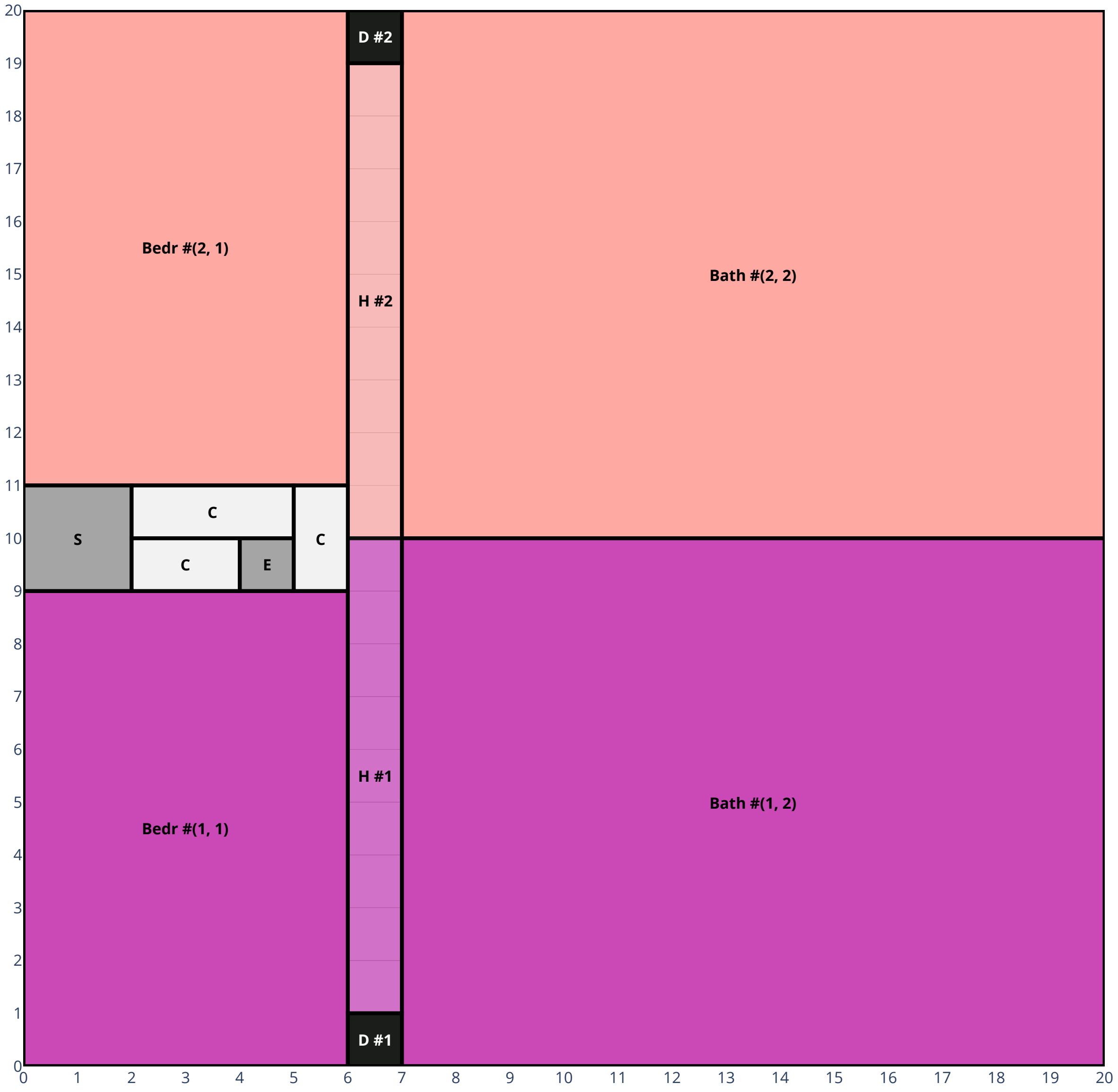
```
In [21]: visualize_solution(solutions, 1)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



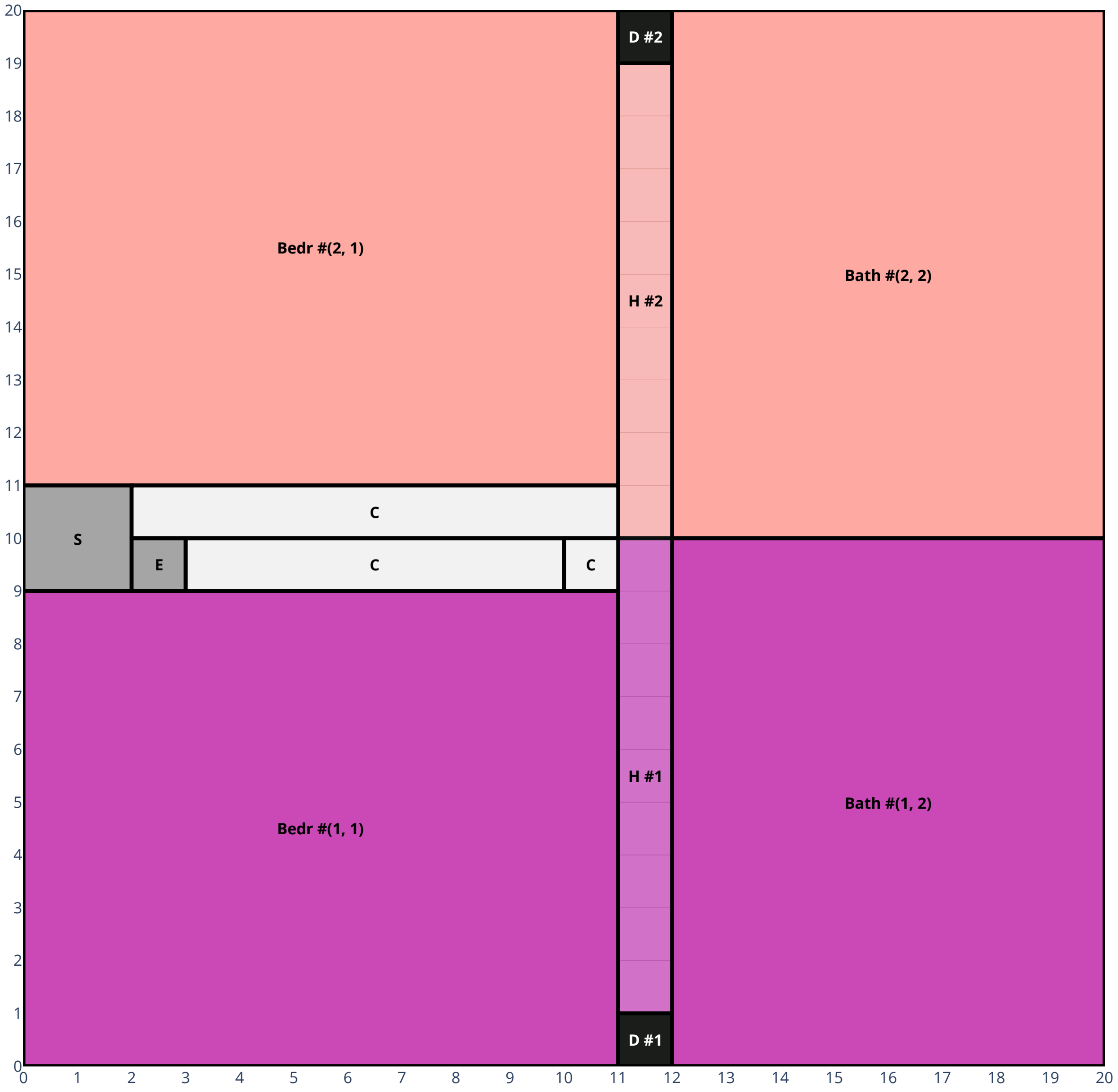
```
In [22]: visualize_solution(solutions, 50)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



```
In [23]: visualize_solution(solutions, 100)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



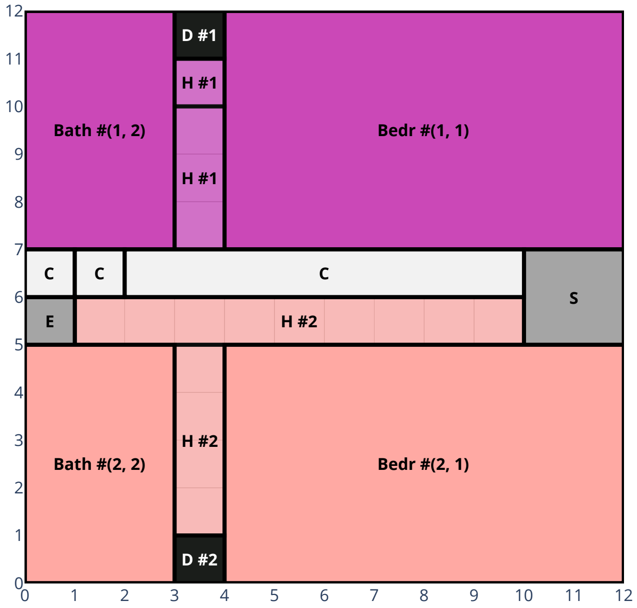
Input 6

```
In [24]: solutions = test_run('inputs/floor_input_6.json', 100)
```

```
Started search...
OPTIMAL
Number of solutions is: 100
Done in 16.9 seconds
```

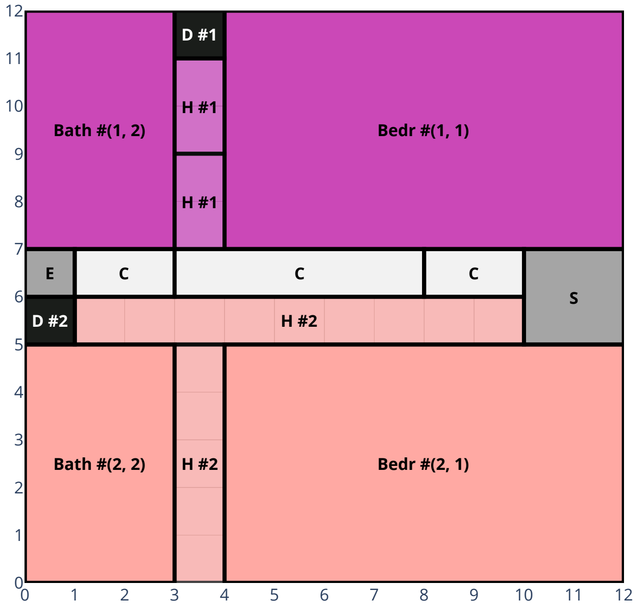
In [25]:
```
visualize_solution(solutions, 1)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



In [26]:
```
visualize_solution(solutions, 50)
```

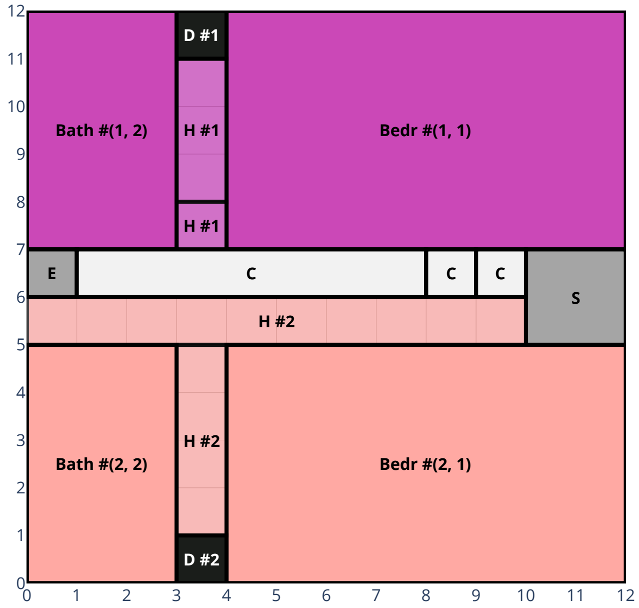
```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```



In [27]:

```
visualize_solution(solutions, 100)
```

```
Soft Constraints:
-----------------
Soft constraint allRoomsHaveDaylight: True
Soft constraint allRoomsHaveGoodSpacings: True
Soft constraint allBedroomsAreClose: True
Soft constraint mainBathroomIsClose: True
```


D #1
Bath #(1, 2)
H #1
Bedr #(1, 1)
H #1
E
C
C
C
S
H #2
H #2
Bath #(2, 2)
Bedr #(2, 1)
D #2
12
11
10
9
8
7
6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8
9
10
11
12