

Pseudocode

Program begins:

Create a list of instances from “instance01.txt” to “instance99.txt”; ----- $O(99)$

Create a list of solutions from “solution01.txt” to “solution99.txt”; ----- $O(99)$

For each instance:

// Find all pairs.

// Find all horizontal pairs.

For $i = 1$ to n , do: ----- $O(n^2)$

 For $j = i+1$ to n , do:

 horizontal_pairs.append([i, j])

 Endfor

Endfor

// Find all vertical pairs.

For $i = 0$ to $n-1$, do: ----- $O(n^2)$

 For $j = i$ to $n-1$, do:

 vertical_pairs.append([i, j]);

 Endfor

Endfor

// Find all lines.

For $i = 1$ to $n-1$, do: ----- $O(n)$

 lines.append(['v', $i + 0.5$]);

Endfor

For $i = 1$ to $n-1$, do: ----- $O(n)$

 lines.append(['h', $i + 0.5$]);

Endfor

// Form sets using all pairs and lines.

For each line in lines, do: ----- $O(2n-2) * O(n * (n-1) / 2)$

 For each pair in all pairs, do: $= O(n) * O(n^2)$

 If line separates pair, then: $= O(n^3)$

 set.append(pair);

 Endif

 Endfor

 sets.append(set);

Endfor

// Do the greedy procedure described in Section 35.3 of the textbook.

While |pairs| > 0, do: ----- $O(n^2) * 3 * O(n)$

// Find the largest sets. $= O(n^3)$

 For each set in sets, do: ----- $O(n)$

 max_count = max(max_count, |set|)

 Endfor

 For each set in sets, do: ----- $O(n)$

 If |set| == max_count, then:

 largest_sets.append(set)

 Endif

 Endfor

 Randomly pick one from largest_sets;

 Add the line that separates this randomly picked set to result;

```

// Eliminate the elements of this randomly picked set in pairs and all other sets.
For each pair in the randomly picked set, do: -----O(1)
    Delete this pair in pairs;
Endfor
For each pair in the randomly picked set, do: -----O(n)
    For each set in all other sets, do:
        If this pair appears in the set, then:
            Delete the pair from the set;
        Endfor
    Endfor
Endwhile
Endfor
Save result to a "solution#.txt", where # is equal to file number related to current iteration;
Program ends

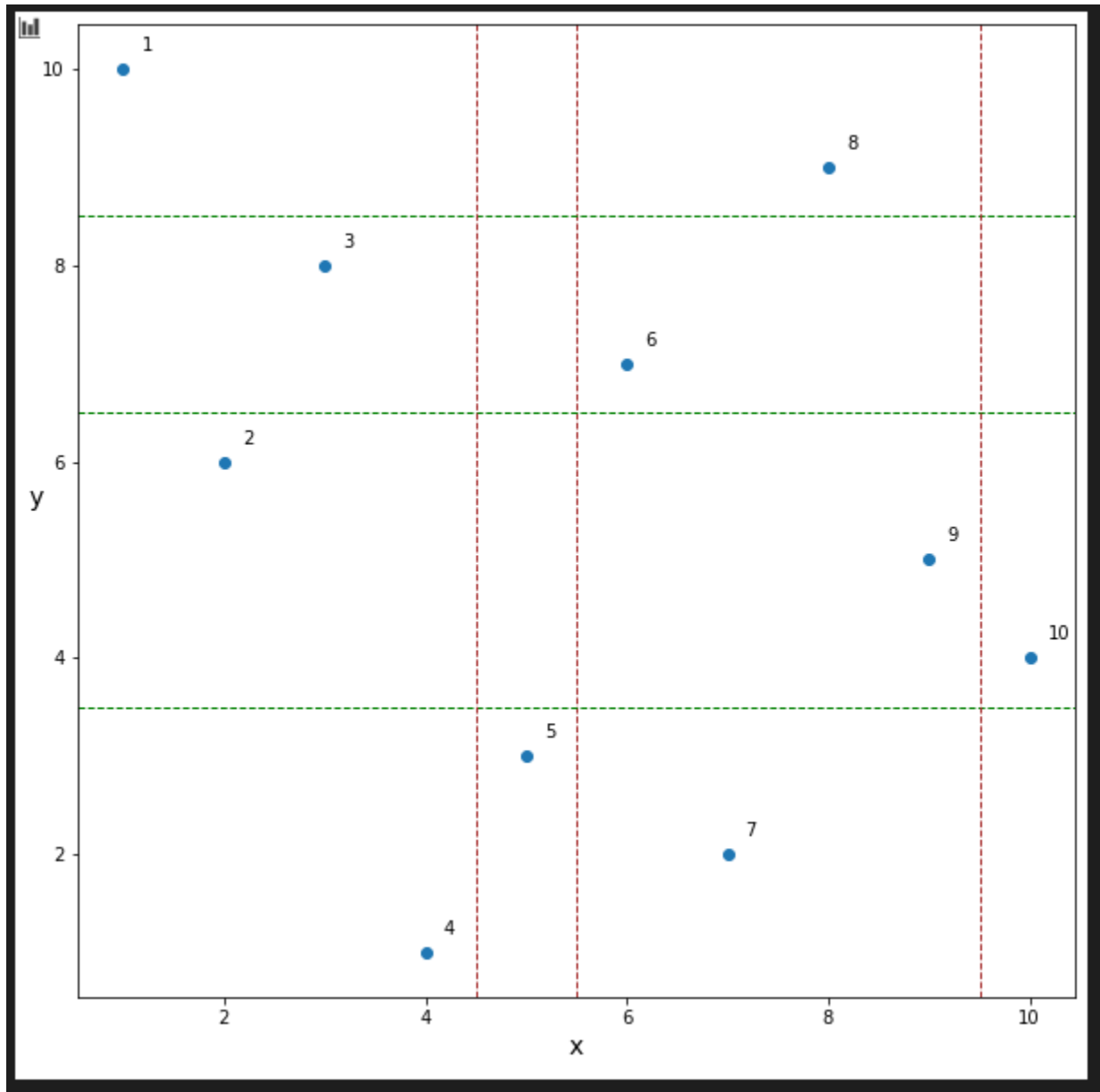
```

Time complexity:

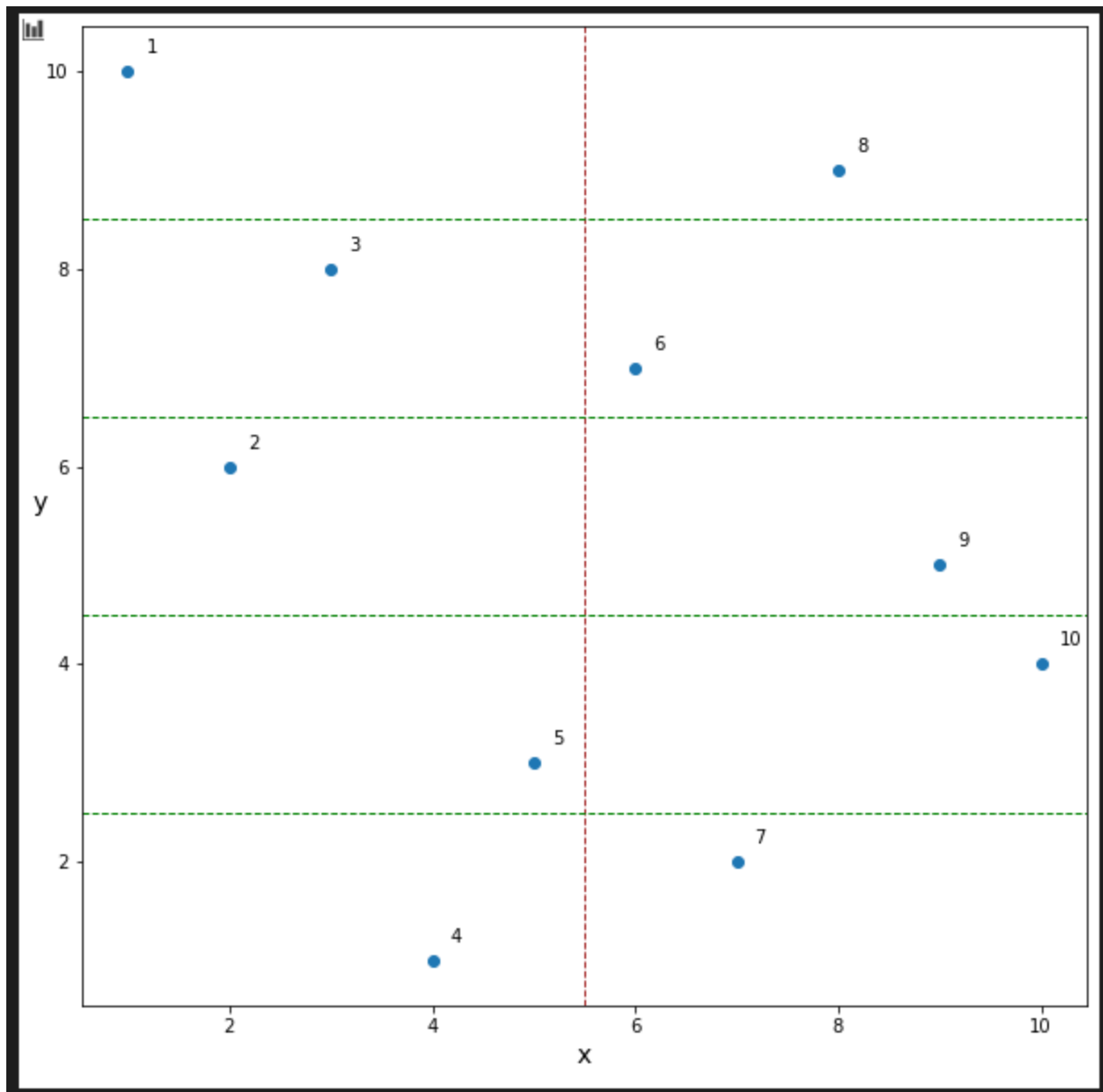
The number of points is n ; based on n , we can calculate the total number of lines which is $2*n-2$ and the number of all pairs which is $n*(n-1)/2$. The number of sets is equal to the number of lines which is also $2*n-2$. And we have in total 99 instances, so the outmost for loop will run 99 times; inside the for loop, all the loops sum up and simplify to just $O(n^3)$. Therefore, eventually the time complexity for this algorithm will be $99 * O(n^3)$ which is just **$O(n^3)$** .

A better solution (or randomness of the program):

On instance03, the algorithm sometimes generates:



There are 6 lines generated. However, we can do it with 5 lines like this:



However, this solution sometimes can also be generated by the program because there is a random selection process (see the line marked **red** above in the greedy section) kicking in during the running of the greedy procedure where there are multiple largest sets, from which one largest set will be randomly picked. So different selections at this stage will lead to different selection paths after, therefore lead to different final results.

How the algorithm runs on the instance (instance03):

Iteration1-----

Current pairs: [[1, 2], [1, 3], [1, 4], [1, 5], [1, 6], [1, 7], [1, 8], [1, 9], [1, 10], [2, 3], [2, 4], [2, 5], [2, 6], [2, 7], [2, 8], [2, 9], [2, 10], [3, 4], [3, 5], [3, 6], [3, 7], [3, 8], [3, 9], [3, 10], [4, 5], [4, 6], [4, 7], [4, 8], [4, 9], [4, 10], [5, 6], [5, 7], [5, 8], [5, 9], [5, 10], [6, 7], [6, 8], [6, 9], [6, 10], [7, 8], [7, 9], [7, 10], [8, 9], [8, 10], [9, 10]]

Current sets:

S1 = [[1, 2], [1, 3], [1, 4], [1, 5], [1, 6], [1, 7], [1, 8], [1, 9], [1, 10]]

S2 = [[1, 3], [1, 4], [1, 5], [1, 6], [1, 7], [1, 8], [1, 9], [1, 10], [2, 3], [2, 4], [2, 5], [2, 6], [2, 7], [2, 8], [2, 9], [2, 10]]

S3 = [[1, 4], [1, 5], [1, 6], [1, 7], [1, 8], [1, 9], [1, 10], [2, 4], [2, 5], [2, 6], [2, 7], [2, 8], [2, 9], [2, 10], [3, 4], [3, 5], [3, 6], [3, 7], [3, 8], [3, 9], [3, 10]]

S4 = [[1, 5], [1, 6], [1, 7], [1, 8], [1, 9], [1, 10], [2, 5], [2, 6], [2, 7], [2, 8], [2, 9], [2, 10], [3, 5], [3, 6], [3, 7], [3, 8], [3, 9], [3, 10], [4, 5], [4, 6], [4, 7], [4, 8], [4, 9], [4, 10]]

S5 = [[1, 6], [1, 7], [1, 8], [1, 9], [1, 10], [2, 6], [2, 7], [2, 8], [2, 9], [2, 10], [3, 6], [3, 7], [3, 8], [3, 9], [3, 10], [4, 6], [4, 7], [4, 8], [4, 9], [4, 10], [5, 6], [5, 7], [5, 8], [5, 9], [5, 10]]

S6 = [[1, 7], [1, 8], [1, 9], [1, 10], [2, 7], [2, 8], [2, 9], [2, 10], [3, 7], [3, 8], [3, 9], [3, 10], [4, 7], [4, 8], [4, 9], [4, 10], [5, 7], [5, 8], [5, 9], [5, 10], [6, 7], [6, 8], [6, 9], [6, 10]]

S7 = [[1, 8], [1, 9], [1, 10], [2, 8], [2, 9], [2, 10], [3, 8], [3, 9], [3, 10], [4, 8], [4, 9], [4, 10], [5, 8], [5, 9], [5, 10], [6, 8], [6, 9], [6, 10], [7, 8], [7, 9], [7, 10]]

S8 = [[1, 9], [1, 10], [2, 9], [2, 10], [3, 9], [3, 10], [4, 9], [4, 10], [5, 9], [5, 10], [6, 9], [6, 10], [7, 9], [7, 10], [8, 9], [8, 10]]

S9 = [[1, 10], [2, 10], [3, 10], [4, 10], [5, 10], [6, 10], [7, 10], [8, 10], [9, 10]]

S10 = [[1, 4], [2, 4], [3, 4], [4, 5], [4, 6], [4, 7], [4, 8], [4, 9], [4, 10]]

S11 = [[1, 4], [1, 7], [2, 4], [2, 7], [3, 4], [3, 7], [4, 5], [4, 6], [4, 8], [4, 9], [4, 10], [5, 7], [6, 7], [7, 8], [7, 9], [7, 10]]

S12 = [[1, 4], [1, 5], [1, 7], [2, 4], [2, 5], [2, 7], [3, 4], [3, 5], [3, 7], [4, 6], [4, 8], [4, 9], [4, 10], [5, 6], [5, 8], [5, 9], [5, 10], [6, 7], [7, 8], [7, 9], [7, 10]]

S13 = [[1, 4], [1, 5], [1, 7], [1, 10], [2, 4], [2, 5], [2, 7], [2, 10], [3, 4], [3, 5], [3, 7], [3, 10], [4, 6], [4, 8], [4, 9], [5, 6], [5, 8], [5, 9], [6, 7], [6, 10], [7, 8], [7, 9], [8, 10], [9, 10]]

S14 = [[1, 4], [1, 5], [1, 7], [1, 9], [1, 10], [2, 4], [2, 5], [2, 7], [2, 9], [2, 10], [3, 4], [3, 5], [3, 7], [3, 9], [3, 10], [4, 6], [4, 8], [5, 6], [5, 8], [6, 7], [6, 9], [6, 10], [7, 8], [8, 9], [8, 10]]

S15 = [[1, 2], [1, 4], [1, 5], [1, 7], [1, 9], [1, 10], [2, 3], [2, 6], [2, 8], [3, 4], [3, 5], [3, 7], [3, 9], [3, 10], [4, 6], [4, 8], [5, 6], [5, 8], [6, 7], [6, 9], [6, 10], [7, 8], [8, 9], [8, 10]]

S16 = [[1, 2], [1, 4], [1, 5], [1, 6], [1, 7], [1, 9], [1, 10], [2, 3], [2, 8], [3, 4], [3, 5], [3, 6], [3, 7], [3, 9], [3, 10], [4, 8], [5, 8], [6, 8], [7, 8], [8, 9], [8, 10]]

S17 = [[1, 2], [1, 3], [1, 4], [1, 5], [1, 6], [1, 7], [1, 9], [1, 10], [2, 8], [3, 8], [4, 8], [5, 8], [6, 8], [7, 8], [8, 9], [8, 10]]

S18 = [[1, 2], [1, 3], [1, 4], [1, 5], [1, 6], [1, 7], [1, 8], [1, 9], [1, 10]]

Largest_sets: [4, 13] (*The elements here are indices in the sets list.*)

Pick: 4 (*We pick the 5th set, which is S5 above shown in red, also shown below.*)

Picked_set: [[1, 6], [1, 7], [1, 8], [1, 9], [1, 10], [2, 6], [2, 7], [2, 8], [2, 9], [2, 10], [3, 6], [3, 7], [3, 8], [3, 9], [3, 10], [4, 6], [4, 7], [4, 8], [4, 9], [4, 10], [5, 6], [5, 7], [5, 8], [5, 9], [5, 10]]

Picked_line: ['v', 5.5] (*This is the line that separates all the pairs in the picked set, so at the end of this iteration we add this line to our result list. For the next iteration, the elements in the picked set will be deleted from the pairs pool and from the other sets that originally have these elements.*)

Iteration2-----

Current pairs: [[1, 2], [1, 3], [1, 4], [1, 5], [2, 3], [2, 4], [2, 5], [3, 4], [3, 5], [4, 5], [6, 7], [6, 8], [6, 9], [6, 10], [7, 8], [7, 9], [7, 10], [8, 9], [8, 10], [9, 10]] (*So as we can see, after the first iteration where we picked a very large set, a lot of pairs have been removed and so now the pool for pairs is a lot smaller.*)

Current sets:

S1 = [[1, 2], [1, 3], [1, 4], [1, 5]]

S2 = [[1, 3], [1, 4], [1, 5], [2, 3], [2, 4], [2, 5]]

S3 = [[1, 4], [1, 5], [2, 4], [2, 5], [3, 4], [3, 5]]

S4 = [[1, 5], [2, 5], [3, 5], [4, 5]]

S5 = [[]] S6 = [[6, 7], [6, 8], [6, 9], [6, 10]]

S7 = [[6, 8], [6, 9], [6, 10], [7, 8], [7, 9], [7, 10]]

S8 = [[6, 9], [6, 10], [7, 9], [7, 10], [8, 9], [8, 10]]

S9 = [[6, 10], [7, 10], [8, 10], [9, 10]]

S10 = [[1, 4], [2, 4], [3, 4], [4, 5]]

S11 = [[1, 4], [2, 4], [3, 4], [4, 5], [6, 7], [7, 8], [7, 9], [7, 10]] S12 = [[1, 4], [1, 5], [2, 4], [2, 5], [3, 4], [3, 5], [6, 7], [7, 8], [7, 9], [7, 10]]

S13 = [[1, 4], [1, 5], [2, 4], [2, 5], [3, 4], [3, 5], [6, 7], [6, 10], [7, 8], [7, 9], [8, 10], [9, 10]]

S14 = [[1, 4], [1, 5], [2, 4], [2, 5], [3, 4], [3, 5], [6, 7], [6, 9], [6, 10], [7, 8], [8, 9], [8, 10]]

S15 = [[1, 2], [1, 4], [1, 5], [2, 3], [3, 4], [3, 5], [6, 7], [6, 9], [6, 10], [7, 8], [8, 9], [8, 10]]

S16 = [[1, 2], [1, 4], [1, 5], [2, 3], [3, 4], [3, 5], [6, 8], [7, 8], [8, 9], [8, 10]]

S17 = [[1, 2], [1, 3], [1, 4], [1, 5], [6, 8], [7, 8], [8, 9], [8, 10]] S18 = [[1, 2], [1, 3], [1, 4], [1, 5]]

Largest_sets: [12, 13, 14]

Pick: 14

Picked_set: [[1, 2], [1, 4], [1, 5], [2, 3], [3, 4], [3, 5], [6, 7], [6, 9], [6, 10], [7, 8], [8, 9], [8, 10]]

Picked_line: ['h', 6.5]

Iteration3-----

Current pairs: [[1, 3], [2, 4], [2, 5], [4, 5], [6, 8], [7, 9], [7, 10], [9, 10]]

Current sets:

S1 = [[1, 3]]

S2 = [[1, 3], [2, 4], [2, 5]]

S3 = [[2, 4], [2, 5]]

S4 = [[2, 5], [4, 5]]

S5 = [[]]

S6 = [[6, 8]]

S7 = [[6, 8], [7, 9], [7, 10]]

S8 = [[7, 9], [7, 10]]

S9 = [[7, 10], [9, 10]]

S10 = [[2, 4], [4, 5]]

S11 = [[2, 4], [4, 5], [7, 9], [7, 10]]

S12 = [[2, 4], [2, 5], [7, 9], [7, 10]]

S13 = [[2, 4], [2, 5], [7, 9], [9, 10]]

S14 = [[2, 4], [2, 5]]

S15 = [[]]

S16 = [[6, 8]]

```
S17 = [[1, 3], [6, 8]]
```

```
S18 = [[1, 3]]
```

```
Largest_sets: [10, 11, 12]
```

```
Pick: 11
```

```
Picked_set: [[2, 4], [2, 5], [7, 9], [7, 10]]
```

```
Picked_line: ['h', 3.5]
```

Iteration4-----

```
Current pairs: [[1, 3], [4, 5], [6, 8], [9, 10]]
```

```
Current sets:
```

```
S1 = [[1, 3]]
```

```
S2 = [[1, 3]]
```

```
S3 = [[]]
```

```
S4 = [[4, 5]]
```

```
S5 = [[]]
```

```
S6 = [[6, 8]]
```

```
S7 = [[6, 8]]
```

```
S8 = [[]]
```

```
S9 = [[9, 10]]
```

```
S10 = [[4, 5]]
```

```
S11 = [[4, 5]]
```

```
S12 = [[]]
```

```
S13 = [[9, 10]]
```

```
S14 = [[]]
```

```
S15 = [[]]
```

```
S16 = [[6, 8]]
```

```
S17 = [[1, 3], [6, 8]]
```

```
S18 = [[1, 3]]
```

```
Largest_sets: [16]
```

```
Pick: 16
```

```
Picked_set: [[1, 3], [6, 8]]
```

```
Picked_line: ['h', 8.5]
```

Iteration5-----

```
Current pairs: [[4, 5], [9, 10]]
```

```
Current sets:
```

```
S1 = [[]]
```

```
S2 = [[]]
```

```
S3 = [[]]
```

```
S4 = [[4, 5]]
```

```
S5 = [[]]
```

```
S6 = [[]]
```

```
S7 = [[]]
```

```
S8 = [[]]
```

```
S9 = [[9, 10]]
```

```
S10 = [[4, 5]]
```

```
S11 = [[4, 5]]
```

```
S12 = [[]]
```

```
S13 = [[9, 10]]
```

```
S14 = [[]]
```

```
S15 = [[]]
```

```
S16 = [[]]
```

```
S17 = [[]]
```

```
S18 = [[]]
```

Largest_sets: [3, 8, 9, 10, 12]

Pick: 8

Picked_set: [[9, 10]]

Picked_line: ['v', 9.5]

Iteration6-----

Current pairs: [[4, 5]]

Current sets:

S1 = [[]]

S2 = [[]]

S3 = [[]]

S4 = [[4, 5]]

S5 = [[]]

S6 = [[]]

S7 = [[]]

S8 = [[]]

S9 = [[]]

S10 = [[4, 5]]

S11 = [[4, 5]]

S12 = [[]]

S13 = [[]]

S14 = [[]]

S15 = [[]]

S16 = [[]]

S17 = [[]]

S18 = [[]]

Largest_sets: [3, 9, 10]

Pick: 3

Picked_set: [[4, 5]]

Picked_line: ['v', 4.5]

Output -----

Result: [['v', 5.5], ['h', 6.5], ['h', 3.5], ['h', 8.5], ['v', 9.5], ['v', 4.5]]