

Homework Sheet 3

Authors

Abdullah Oğuz Topçuoğlu
Ahmed Waleed Ahmed Badawy Shora
Yousef Mostafa Farouk Farag

Tutors

Maryna Dernovaia
Jan-Hendrik Gindorf
Thorben Johr

Exercise 3

We can do something similar to LsdRadix sort we saw in the lecture.

Pseudocode:

```
function SortGridPoints(A[1..n]) {  
    redefine key(point) := point.y  
    CountingSort(A)  
  
    redefine key(point) := point.x  
    CountingSort(A)  
}
```

Correctness:

- The first CountingSort sorts the points by their y coordinates.
- The second CountingSort sorts the points by their x coordinates, but since CountingSort is stable, the order of points with the same x coordinate is preserved.
- Therefore after both sorts the points are sorted lexicographically by (x, y).
- That's also what we did in the lecture for LSDRadixSort we started from the least significant digit to the most significant digit. The same idea applies here.

Running Time Analysis:

- Each CountingSort runs in time $O(n + k)$ where k is the range of the keys.
- Here the keys are the x and y coordinates of the points.
- Since the points are connected, the range of x coordinates is at most n and the range of y coordinates is also at most n .
- Therefore each CountingSort runs in time $O(n + n) = O(n)$.
- Since we perform two CountingSorts, the total running time is $O(n) + O(n) = O(n)$.