

CSCI432 GW6

Brock Ellefson, Seth Severa

November 28, 2017

1 Diversify your group

We talked about this earlier, we will diversify for the next assignment.

2 Recall the Skyline problem. Give pseudocode for the sweepline algorithm

SkyLine sweep-line algorithm

Algorithm 1 Skyline

```
1: List buildings
2: PriorityQueue events
3: while currentBuilding  $\neq$  null do
4:   events.add(currentBuilding, currentBuilding.left)
5:   events.add(currentBuilding, currentBuilding.right)
6:   currentBuilding = currentBuilding.next
7: end while
8:
9: List skyline
10: List currSkyline
11: double maxHeight = 0
12: while events.peek()  $\neq$  null do
13:   if event.peek().isLeft() then
14:     currSkyline.add(event.peek())
15:     if maxheight  $\neq$  checkMaxHeight(currSkyline) then sky-
line.add(event.pop().left, checkmaxHeight(currSkyline))
16:   end if
17:   else
18:     currSkyline.remove(event.peek())
19:     if maxheight  $\neq$  checkMaxHeight(currSkyline) then sky-
line.add(event.pop().right, checkmaxHeight(currSkyline))
20:   end if
21: end if
22: end while
23: return skyline
```

The loop invariant of this algorithm is that as the sweepline goes from left to right we always use the height of the current tallest building.

3 For the sweepline algorithm that computes whether there exist two line segments that intersect, explain how a line segment is added to or removed from T (the data structure that keeps track of the lines that intersect the sweep line).

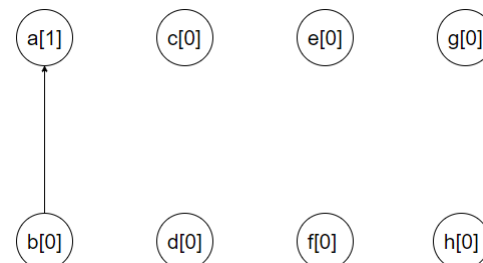
T is a red-black tree, we sweep through the graph from left to right, stopping at endpoints to check for intersections and add or remove line segments to T . Every time we stop at an endpoint, we check the order of T , if the order of T is different than the order of T from the previous endpoint we stopped at, then there was an intersection. After this check, if the endpoint of the line segment the sweepline is on is the leftmost endpoint of the line, then we add that line segment to the red-black tree. If the endpoint of the line segment the sweepline is on is the rightmost endpoint of the line, then we remove that line segment from the red-black tree.

4 Union and Path Compression

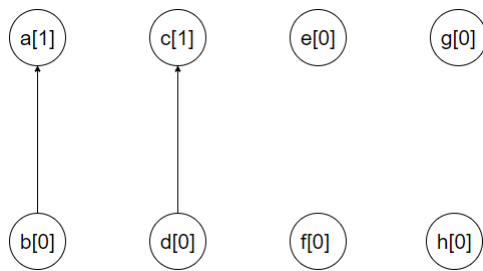
This is our beginning, a total of 8 components all of rank 0



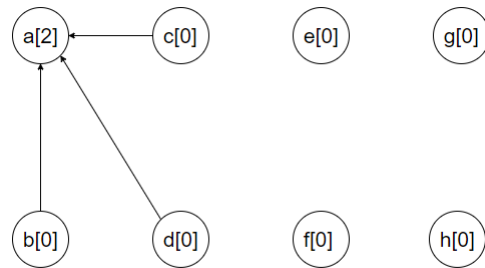
This iteration combines two components of equal rank. Now a has a rank of one, and there are 7 components



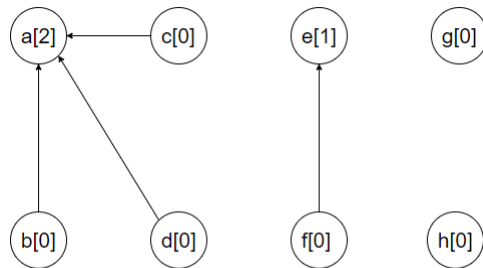
This iteration combines two components of equal rank. Now a and c have a rank of one, and there are 6 components



This iteration adds the two components together, since a and c components both have the same rank, either could be the parent of the combined component, so we choose a. A's rank gets increased by 1, then we use path compression, and have d point to a as well. We now have 5 components.



This iteration combines two components of equal rank. Now e has a rank of one. We now have 4 components.



This iteration combines two components of different rank. Since e has a higher rank than g, e will become g's parent. E's rank will not change. We now only have 3 components

