CS4800: Algorithms — S'18 — Jonathan Ullman

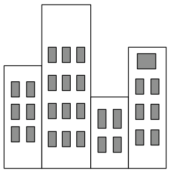
Homework 3
Due Friday February 2 at 11:59pm via Gradescope

Name:
Collaborators:

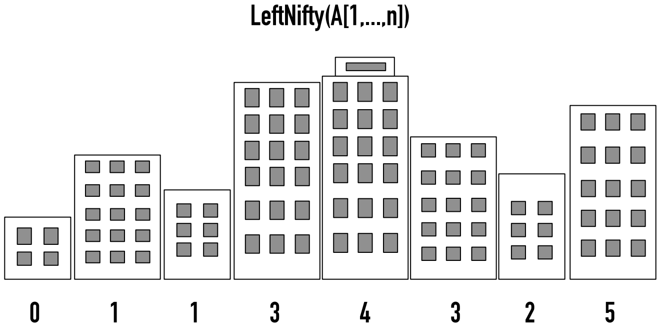
- Make sure to put your name on the first page. If you are using the LaTeX template we provided, then you can make sure it appears by filling in the `yourname` command.
- This assignment is due Friday February 2 at 11:59pm via Gradescope. No late assignments will be accepted. Make sure to submit something before the deadline.
- Solutions must be typeset in LaTeX. If you need to draw any diagrams, you may draw them by hand as long as they are embedded in the PDF. I recommend using the source file for this assignment to get started.
- I encourage you to work with your classmates on the homework problems. *If you do collaborate, you must write all solutions by yourself, in your own words.* Do not submit anything you cannot explain. Please list all your collaborators in your solution for each problem by filling in the `yourcollaborators` command.
- Finding solutions to homework problems on the web, or by asking students not enrolled in the class is strictly prohibited.

Problem 1. *Skylines*

A great skyline must have variations. Define the left skyline function, denoted $n_L(s)$, of a skyline s as the total number of times that a building is taller than one of its left neighbors. The right skyline function, $n_R(s)$, is defined analogously. The skyline s below has 4 buildings, and $n_L(s) = 3$ (because building 2 is taller than building 1 and building 4 is taller than buildings 3 and 1) and $n_R(s) = 3$ (because building 1 is taller than building 3 and building 2 is taller than buildings 3 and 4).



As another example, this skyline has $n_L(s) = 19$, the contribution of each building is listed below the building.



(a) Design and analyze a divide-and-conquer algorithm that computes the left and right skyline functions of a skyline s with n buildings. The input $A[1,\ldots,n]$ consists of the heights of each building along a street in left to right order; assume all buildings have unique heights. Your solution should have a running time of $\Theta(n \log n)$. Write pseudocode for your algorithm, justify its correctness, and analyze its running time.[1]

Solution:

(b) In this part, you will implement your skyline algorithm and test it on real data. Register and take on the challenge at `https://www.hackerrank.com/contests/cs4800-s18-ullman`. Please write your username and score here.

Solution:

[1] Hint: Does this problem sound familiar?