CMSC 27200 - Winter 2021 — Andrew Drucker, Lorenzo Orecchia

due tuesday, january 26 at 12 noon via gradescope

# H2.Hamlet

- The assignment is due at Gradescope on Tuesday, January 26 at 12 noon.
- You can either type your homework using LaTex or scan your handwritten work. We will provide a LaTex template for each homework. If you writing by hand, please fill in the solutions in this template, inserting additional sheets as necessary. This will facilitate the grading.
- You are permitted to study and discuss the problems with 2 other students (per problem; any section). However, *you must write up your own solutions, in your own words*. Do not submit anything you cannot explain. If you do collaborate with any of the other students on any problem, please do list all your collaborators in your submission for each problem.
- Similarly, please list any other source you have used for each problem, including other textbooks or websites.
- *Show your work.* Answers without justification will be given little credit.

PROBLEM 1 (25 POINTS) *Solve exercise 3 in Chapter 4 in the Kleinberg-Tardos textbook. (trucking)*

**Solution:** Your solution goes here.

Extra Space for your solution

PROBLEM 2 (25 POINTS) *Solve exercise 5 in Chapter 4 in the Kleinberg-Tardos textbook. (cell phone towers)*

*Please note! Here and elsewhere, when the authors (or your instructors) say "give an algorithm" without further instructions, they mean "give an algorithm AND prove that it is correct and runs in polynomial time". This should be assumed henceforth.*

**Solution:** Your solution goes here.

Extra Space for your solution

PROBLEM 3 (25 POINTS) *The Running Sums problem is defined as follows:*

***Input:*** *a sequence* $(a_1, \ldots, a_n)$*, where each* $a_i$ *is either* 1 *or* -1*.*

***Desired output:*** *a sequence* $(b_1, \ldots, b_n)$*, where each* $b_i$ *is either* 0 *or* 1*.*
*Your goal is to* ***minimize*** $\sum_{1 \le i \le n} b_i$*, subject to the* ***constraint*** *that*

$$\sum_{1 \le i \le j} (a_i + b_i) \;\ge\; 0\,, \qquad \text{for all} \quad j \in \{1, 2, \ldots, n\}\,.$$

*Give an algorithm for this problem that, for full credit, should run in* $O(n)$ *steps.*

**Solution:** Your solution goes here.

Extra Space for your solution

PROBLEM 4 (25 POINTS) *In the Hopping Game, there is a sequence of n spaces. You begin at space 0 and at each step, you can hop 1,2,3, or 4 spaces forward. However, some of the spaces have obstacles and if you land on an obstacle, you lose.*

*Give a greedy algorithm which, given an array* $A[1, \ldots, n-1]$ *of Boolean values with* $A[i]$ *indicating the presence/absence of obstacle at position* $i \in [1, n-1]$*, find the minimum number of hops needed to reach space n without losing, if it is possible to do so. (We assume spaces 0 and n are obstacle-free, and are not part of the input.) Prove that your algorithm is correct. For full credit, your algorithm should run in time* $O(n)$*.*

Extra Space for your solution