6.5440: Algorithmic Lower Bounds, Fall 2023

Prof. Erik Demaine, Josh Brunner, Lily Chung, Jenny Diomidova

Problem Set 2 Solution

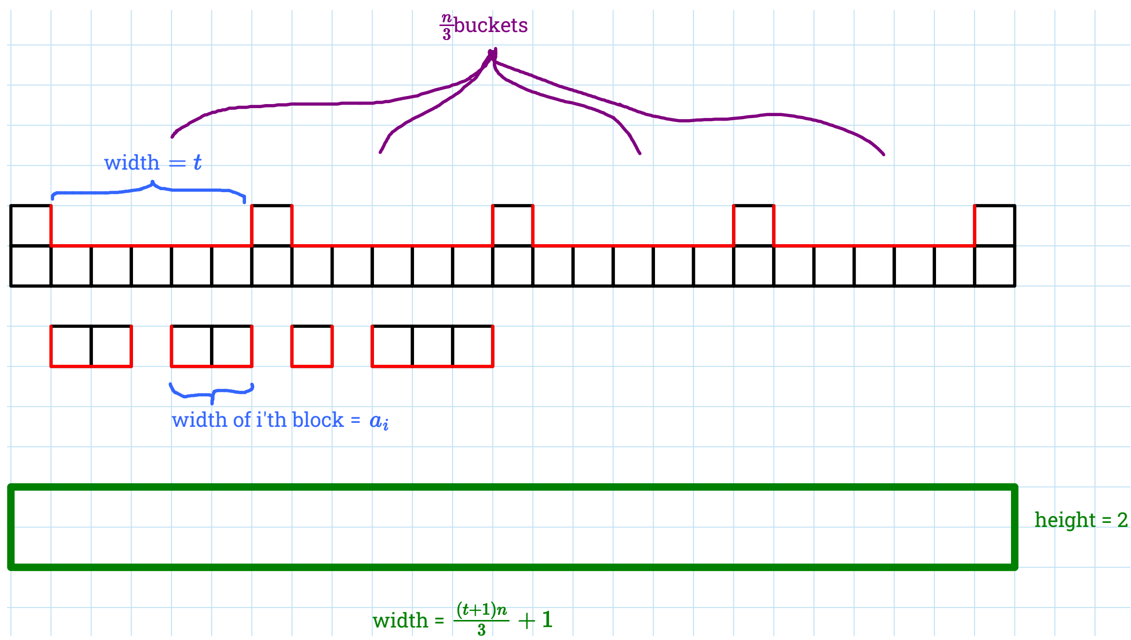
Due: Monday, September 18, 2023 at noon

Problem 2.1 [2×n edge-matching]. In Lecture 2, we saw a reduction from 3-Partition to edge-matching puzzles. Recall that an (unsigned) ***edge-matching puzzle*** consists of mn tiles, where each tile is a unit square whose sides are each labeled with a color. The goal is to place the mn given tiles into an $m \times n$ rectangle such that tiles match in color on shared edges.

Give a reduction from 3-Partition to show that edge-matching puzzles are NP-hard even when the following two conditions hold simultaneously:

- $m = 2$, i.e., the puzzle is $2 \times n$; and
- tiles cannot be rotated or flipped (only translated).

Solution:



We reduce from 3-Partition. We are given an instance $A = \{a_1, \ldots, a_n\}$ with target sum t, where each $a_i \in \left(\frac{t}{4}, \frac{t}{2}\right)$. The a_i and t are polynomial-size.

See the figure. Red edges are all the same color, but each black edge in the figure represents a single unique color. Each such color appears either twice or once depending on whether the edge in the figure is incident to two tiles or only one. Since the number of colors appearing once is equal to the perimeter of the $\left(\frac{(t+1)n}{3} + 1\right) \times 2$ target box, no other edges can be placed on the boundary. This ensures that the tile edges bearing the colors appearing twice must be paired up with each other, forming the shapes shown.

There is a single large shape which forms $\frac{n}{3}$ buckets each of width t, as well as n small blocks, the i'th of which is an $a_i \times 1$ rectangle. This instance can be computed in polynomial time because t and a_i are polynomially bounded.

The instance is solvable if and only if the small blocks can be packed exactly into the buckets. This is true if and only if there exists a solution to the 3-Partition instance, since the assumption that all $a_i \in \left(\frac{t}{4}, \frac{t}{2}\right)$ ensures that each bucket will contain exactly 3 blocks.