

# Problem Set 8

Albert-Kenneth Okine

May 8, 2024

## References

- *Introduction to Algorithms*, fourth edition.
- Jonathon Wang for being my partner

**14-8a**

Suppose that you are given a color picture consisting of an  $m \cdot n$  array  $A[1 : m, 1 : n]$  of pixels, where each pixel specifies a triple of red, green, and blue (RGB) intensities. You want to compress this picture slightly, by removing from each of the  $m$  rows, so that the whole picture becomes one pixel narrower. To avoid possible incongruous visual effects, however, the pixels removed in two adjacent rows must lie in either the same column or adjacent columns. In this way, the pixels removed form a "seam" from the top to the bottom row where successive pixels in the seam are adjacent vertically or diagonally. Show that the number of such possible seams grows at least exponentially in  $m$ , assuming that  $n > 1$ .

Since we want to assume that the number of such possible seams is at least exponentially in  $m$ , assuming that  $n > 1$ , we consider the case where  $n = 2$ . In this case, there are  $n = 2$  columns in the input color picture, meaning that any choice of pixel as the seam in the row will satisfy the same or adjacent column rule. For each pixel in row  $i$ , we can make one of two choices, removing pixel  $A[i, 1]$  or pixel  $A[i, 2]$ . Thus, there are 2 choices per row, with  $n = 2$  rows, meaning there are  $2 \cdot 2 \cdots 2 = 2^m$  unique combinations per  $m$  rows and  $2^m$ .

Any other choice of  $n > 1$  is an extension of this example, and we have proven this property for  $n = 2$ , so we have proven that the number of possible seams grows exponentially in  $m$ , specifically  $2^m$ .

**Designing Algorithm**

Given the following energy function for a pixel, design an algorithm to compute the minimum value vertical seam. i.e. a set of pixels, one per row, where pixels in adjacent rows have to be within one column of each other with minimum total energy. Give the algorithm in pseudocode, give me a proof of correctness, justify the runtime of the algorithm.

$$E(x, y) = \begin{cases} 1000, & (x, y) \text{ is on the border of the image} \\ \sqrt{\sum_{d \in \Delta} d^2}, & \text{otherwise} \end{cases} \quad (1)$$