

## Problem Set 1

Spring 2025

Due: 11:59 pm Monday, 3 Feb

To facilitate grading and timely feedback  
 please note that all submissions are through Gradescope.

Solve each problem on a new page and put your name on each page.  
 Clearly identify all collaborations and resources used in preparing the solutions.  
 You should typeset your solutions using L<sup>A</sup>T<sub>E</sub>X and correctly link  
 solutions on Gradescope.

- Suppose you are in a duel with Sheikh Chilly. You enter the duel arena from the east entrance and he enters from the west and both of you immediately establish eye-contact. Both of you cannot just run directly to each other; instead, you must stay on the path that zig-zags through the arena between the east and west entrances to meet each other. To maintain the proper dramatic tension, the path must be traversed so that you always lie on a direct east-west line.

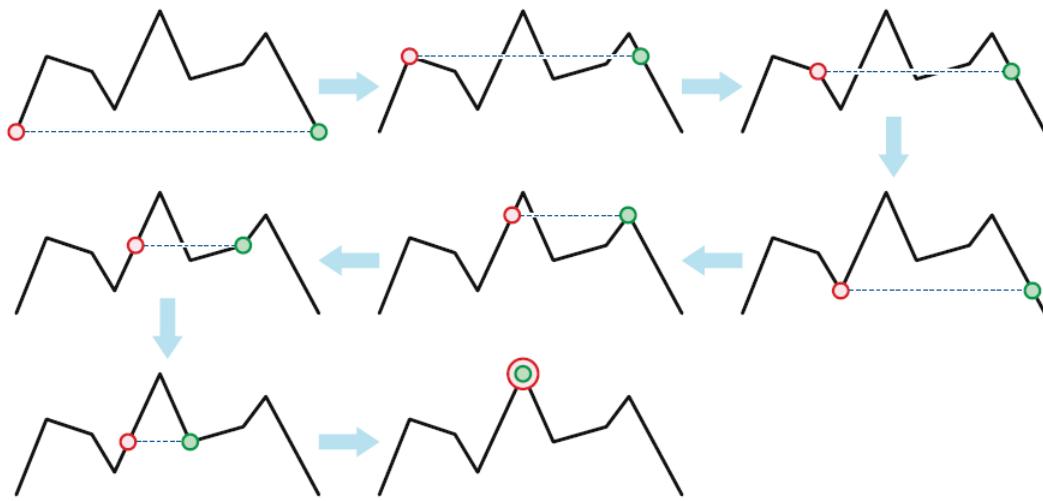


Figure 1: You enter from the east and Sheikh Chilly from the west entrance. You walk backward in step 2, and the Sheikh walks backward in steps 5 and 6.

We can describe the zigzag path as two arrays  $X[0 \dots n]$  and  $Y[0 \dots n]$ , containing the  $x$  and  $y$ -coordinates of the corners of the path, in order from the west entrance to the east entrance. The  $X$  array is sorted in increasing order, and  $Y[0] = Y[n]$ . The path is a sequence of straight line segments connecting these corners.

- Suppose  $Y[0] = Y[n] = 0$  and  $Y[i] > 0$  for every other index  $i$ ; that is, the endpoints of the path are strictly below every other point on the path. Prove that under these conditions, both of you can meet. You may do this by describing a graph that models all

possible locations and transitions for both of you along the path. What are the vertices of this graph? What are the edges? What can you say about the degrees of the vertices?

- (b) If the endpoints of the path are not below every other vertex, both of you may not be able to meet for the duel. Describe an algorithm to decide whether you can meet, without either of you breaking east-west eye contact or stepping off the path, given the arrays  $X[0 \dots n]$  and  $Y[0 \dots n]$  as input. You may do this by building the graph from the previous part. What problem do you need to solve on this graph? What is the algorithm's running time as a function of  $n$ ?
2. Just before the duel you decide to have a hearty meal of aalo-parathas. Assume you are given a pile of  $n$  parathas of different sizes. You want to sort the parathas so that smaller parathas are on top of larger parathas. The only operation you can perform is an *inversion*, that is, for some integer  $k$  between 1 and  $n$ , invert the top  $k$  parathas.

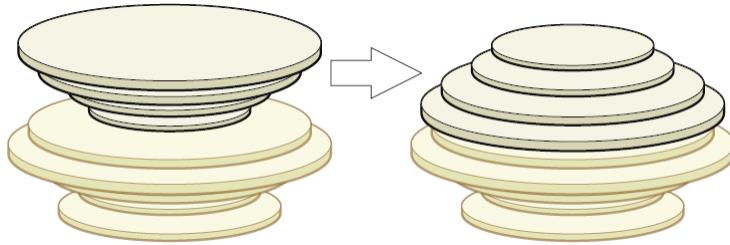


Figure 2: Inverting the top four parathas.

- (a) Describe an algorithm to sort an arbitrary pile of  $n$  parathas using as few inversions as possible. Exactly how many inversions does your algorithm perform in the worst case?
  - (b) Now suppose one side of each paratha is burnt. Describe an algorithm to sort an arbitrary pile of  $n$  parathas, so that the burnt side of every paratha is facing down, using as few inversions as possible. Exactly how many inversions does your algorithm perform in the worst case?
3. Not being able to meet you for the duel in the arena, Sheikh Chilly decides to challenge you in a game. The Sheikh presents you with a complete binary tree with  $4^n$  leaves, with each leaf colored either black or white. There is a token at the root of the tree. To play the game, you and the Sheikh will take turns moving the token from its current node to one of its children. The game will end after  $2n$  moves, when the token lands on a leaf. If the final leaf is black, you lose; if it is white, you win. You move first, so the Sheikh gets the last turn.

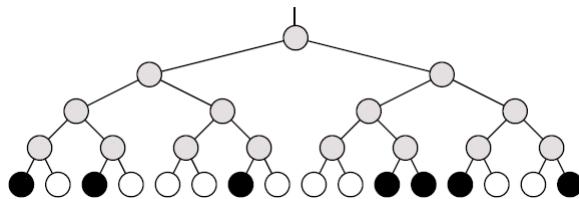


Figure 3: Hint: What logical operation could you evaluate at each node?

- (a) How you can decide whether you should accept the challenge, that is, there exists a winning strategy. What is the complexity of your algorithm?
  - (b) Unfortunately, you do not have enough time to look at every node in the tree. Describe a randomized algorithm that determines whether you can win in  $O(3^n)$  expected time.
4. Consider the following non-standard algorithm for shuffling a deck of  $n$  cards, initially numbered in order from 1 on the top to  $n$  on the bottom. At each step, we remove the top card from the deck and insert it randomly back into in the deck, choosing one of the  $n$  possible positions uniformly at random. The algorithm ends immediately after we pick up card  $n - 1$  and insert it randomly into the deck.
- (a) Prove that this algorithm uniformly shuffles the deck, meaning each permutation of the deck has equal probability. You may do this by proving that at all times, the cards below card  $n - 1$  are uniformly shuffled.
  - (b) What is the exact expected number of steps executed by the algorithm?