**Mini Project – Eitan Bachmat  
Omri Hefez  
Yuval Ratnovsky**

**Gal Shalom**

To determine the points that could be flipped, we implemented the following algorithm:

1. Calculation of Chains and Maximum Weight:  
    Initially, we used a dynamic programming algorithm that takes a list of nodes and determines the longest sequence of nodes with increasing weights, considering their positions. It achieves this by sorting the nodes based on their coordinates, calculating the weight for each node while keeping track of the previous node in the sequence, and then reconstructing the longest increasing weighted sequence. The algorithm returns the list of nodes representing the longest increasing weighted sequence.
2. Weight Allocation Optimization:  
   We employed a randomized approach to select points to add them weight – each time our algorithm seleceted a point in a randomaly manner.  
   For each selected point, the algorithm ran the dynamic programming algorithm described above and found the new max chain weight:  
   if the weight addition surpassed the maximum chain weight, we excluded that specific point from weight increment.  
   To prevent the algorithm from running indefinitely, we imposed a limited number of retries for each run. This ensured that the optimization process did not consume excessive time and resources.  
   The number of retries was dynamically adjusted for each run to strike a balance between optimizing the weight allocation and adhering to practical time constraints.