ED5310 PROJECT REPORT

❖ Details of the students

1. Name: Shaun Mathew
   Roll. No. : CS21B076
2. Name: Chahel Singh
   Roll. No. : CS21B021

➢ Problem Statement:

To find the minimum tunnel width through which our robot can pass.
Due to the complexity of the problem, we could work out the 2D model of the given scenario, but it is also easily scalable to higher dimensions.

➢ Challenges and Solutions:

While working out the solution, we struggled to find relevant online sources that helped us with the problem, so we knew we had to develop something more original. The initial thought was to rotate the robot by a fixed (small) angle and then calculate the minimum width in each instance. This solution is the base version that we applied and goes to O(n*r), n is the number of points, and r is the number of rotations of the body. One way to improve this is to find out the convex hull of the body and then apply rotations so the complexity becomes O(nlogn + k*r), k being the points on the convex hull of the object. After all this thought, we finally came to the most optimised solution to the problem, which had a complexity of O(nlogn). This nlogn is to find the convex hull, and then we solve the problem using two pointers in O(n).
The concept of two pointers is what we took from our approach to merge two independent convex hulls, as taught in class.
The optimised approach takes the convex hull, and then, for each side, it finds the farthest point to it. We keep iterating for each side and maintain a pointer to its most distant point. This pointer to the farthest point does one rotation around the entire hull (throughout the whole algorithm). So the amortised cost for each edge becomes O(1), which gives a total time complexity of O(n) while solving.

➢ Tools:

We have kept the overall process very simple and elegant. The languages used for this project are C++ and Python. We have written the entire code in C++ for the algorithm as it is a faster language. Then for the visualisation part, we used matplotlib.pyplot, which helped us plot and visualise our model's output.

- ➢ Contribution:
  We have been searching for blogs and other relevant details, but finally, what helped us out was the reference materials for the course. After the endsems, when we sat down, we discussed the ideas and the optimised version was developed by a thorough discussion between us. The idea started when Chahel pointed out that we could check each edge and its farthest point. We went on to discuss the perfect optimised implementation for it. I wrote the visualisation codes. We discussed the idea of the naïve and the optimised solution, and I implemented the naïve approach while Chahel implemented the optimised one.

- ➢ Other details:
  The project is easily scalable to 3 dimensions. Due to time constraints and the complexity of the code in 3 dimensions, we skipped through it. A rough idea for the 3d model is to map each edge to a point and keep rotating in that fashion. We also implemented the edge-point method with a higher time complexity of O(n2) for each edge iterate through all the points. But since we optimised it further, we dint include it here.

Link to the drive folder of the project.