



# Drones

Titus Cieslewski

## Medium Task

It turns out there exist areas in which it is risky to fly drones. Still, routers can be anywhere. Your function `init()` is given a map of the risky areas. `minRisk()` will be called up to 1000 times. It will receive the coordinates of a router, and you should return the smallest possible distance on risky territory that needs to be covered by the drone in order to reach that router. Include the return flight – assume a negligible turning radius. Assume that the drone needs to fly directly above the router in order to attack it. Risky zones are defined by "outer" polygons, but they also have holes, which are not risky (inner polygons). There are up to 1000 total polygon edges. Outer polygons only contain inner polygons and inner polygons contain no other polygons. The polygons are non-intersecting and no two given points have the same y-coordinate. There are similar functions as in Easy for data retrieval, and visual debugging functions are available. The demanded accuracy is 0.001 again.

## Tâche Moyenne

Il semblerait qu'il y a des zones de risque, par-dessus lesquelles il est déconseillé de voler. Pourtant, les routers peuvent être partout. On te fournit une carte avec les zones à risque dans `init()`. Après, `minRisk()` va être appelé jusqu'à 1000 fois. Dans cette fonction tu dois calculer la distance minimale sur territoire à risque (le retour inclus), étant donné un seul routeur. Tu peux assumer un rayon de virage négligeable et le drone devra survoler le routeur exactement. Les zones à risque sont définies par des polygones "outer", mais peuvent contenir des trous, sans risqué associé (polygones "inner"). Il y aura au total jusqu'à 1000 arrêtes de polygones. Des polygones "outer" ne peuvent contenir que des polygones "inner" et les dernières ne contient aucun polygone. Il n'y a pas d'intersections entre polygones et chaque point fourni aura une coordonnée y différente. Il y a des fonction d'interrogation des données similaires à ceux dans la tâche facile, et des fonctions de déboguage visuelle sont disponible. La précision demandée est encore 0.001.

## Mittlere Aufgabe

Wie es sich herausstellt gibt es Gebiete in denen es riskant ist mit Drohnen zu fliegen. Doch Router gibt es überall. Zuerst wird `init()` mit der Spezifikation der riskanten Gebiete aufgerufen. Danach folgen bis zu 1000 Anfragen an `minRisk()`. Darin werden dir die Koordinaten eines Routers gegeben, worauf du mit der kleinstmöglichen Distanz, die über riskantem Gebiet zurückgelegt werden muss um an den Router zu kommen (inklusive Rückflug), antworten musst. Du kannst den Wenderadius vernachlässigen und die Drohne muss diesmal direkt über dem Router sein um darauf zuzugreifen. Riskante Zonen sind gegeben als "outer" Polygone, allerdings haben sie auch Löcher über denen risikolos geflogen werden kann ("inner" polygons). Es gibt bis zu 1000 Eckpunkte insgesamt. "Outer" Polygone enthalten nur "inner" Polygone und letztere enthalten selber keine Polygone. Auch gibt es keine Schnittpunkte zwischen Polygonen und keine zwei Router- oder Eckpunkte haben die selbe y-Koordinate. Zum Abfragen der Daten stehen ähnliche Funktionen wie bei der einfachen Aufgabe zur Verfügung, zusammen mit Funktionen zum visuellen Debuggen. Verlangte Genauigkeit ist wieder 0.001.