```
235             process1.terminate()
236             if os.path.exists(recogLineFile) == False:
237                 print('timeout, work not finished.')
238                 print(error)
239 
240             print('second process...\n')
241             existingPoints = readFileOfPoints(recogLineFile)
242             endPoint = existingPoints[0]
243             startPoint = XYPoint([45.0, 60.0])
244             process2 = mp.Process(target=subprocessForSearchingrecogLineOfSi, args=(class1
245             process2.start()
246             existingPoints = []
247             process2.join(timeout=300)
248             process2.terminate()
249 
250         points = readFileOfPoints(recogLineFile)
251         return points
252 
253 
254     def subprocessForSearchingrecogLineOfSi(class1, class2, fileName, initDirection, fileM
255         # define important constants
256         if startPoint is None:
257             meanTotal = (class1.mean + class2.mean) / 2.0
258             currentLocation = XYPoint([ meanTotal[0,0], meanTotal[1,0] ])
259         else:
260             currentLocation = startPoint
261 
262         lastDirection = initDirection
263         recogLinePoints = []
264         criticalPointsOfClass1 = sorted(class1.points, key=currentLocation.distanceFromXYP
265         criticalPointsOfClass2 = sorted(class2.points, key=currentLocation.distanceFromXYP
266 
267         # points for plot completed or not
268         while currentLocation.distanceFromXYPoint(criticalPointsOfClass1[0]) <= 20 or curr
269 
270             for counter in range(int(1.0/STEP)):
271                 currentLocation, lastDirection, isRecogLinePoint = moveToNextLocation(curr
272                 if isRecogLinePoint:
273                     recogLinePoints.append(currentLocation.rawValue)
274 
275             if endPoint != None:
276                 if currentLocation.x <= endPoint.x and currentLocation.y <= endPoint.y:
```