Задача H. Лапта

Автор задачи и разбора - А.Гусаков

Предположим, что мы для заданного времени time умеем находить точку (если она существует), в которую нужно бросить мяч, чтобы до его подъема соперником у нас было в запасе время, не меньшее time. Этим будет заниматься функция TryTime(time: Real; var x_ans, y_ans:Real):boolean. Функция принимает значение False, если такой точки не существует и True, если существует. В случае существования точки, ее абсцисса и ордината будут помещены в x_ans и y_ans соответственно.

Если мы уже реализовали эту финкцию, то наибольшее время мы сможем найти с помощью бинарного поиска. Будем поддерживать такое свойство: искомое максимальное время лежит на отрезке [l, r].

```
l:=0;
r:=MAX_ANS;
  while (r-l>2*EPS) do begin // EPS=точность, с которой мы ищем время
    m:=(l+r)/2; // m=середине отрезка
    if TryTime(m,x_ans,y_ans) then //Если икомое время не меньше m
      l:=m // то отрезок:=его правая половина
    else    // иначе
      r:=m;// отрезок:=его левая половина
  end;
```

Искомым временем будет являться (l+r)/2, а координатами x_ans и y_ans.

Поскольку длина отрезка после каждой итерации цикла уменьшается в 2 раза, то количество вызовов функции TryTime будет достаточно маленьким.

Осталось самое главное – реализовать TryTime(time, x_ans, y_ans). По сути, требуется найти точку, с неотрицательной ординатой, лежащую не дальше, чем D от начала координат, и не принадлежащую ни одному кругу (x[i], y[i], v[i]*time) или определить, что такой точки не существует. Предположим, что такая точка P существует. Тогда будем непрерывно двигать ее вверх, пока во что-нибудь не “упремся”. Если после того как мы во что-то “уперлись” возможно двигаться, повышая высоту, то сделаем это. В итоге, наша точка либо станет самой верхней, то есть (0, D), либо совпадет с точкой пересечения двух окружностей (иначе можно будет двигаться еще выше). Таким образом, чтобы проверить, существует ли наша точка, будем искать ее во множестве точек, состоящем из точек пересечения окружностей (0, 0, D), (x[1], y[1], v[1]*time), … , (x[n], y[n], v[n]*time) и точки (0, D). Проверить, подходит ли нам точка, совсем несложно: надо просто понять, можно ли в нее кидать мяч и что ни один соперник не добежит до нее быстрее, чем за время time.

Всего точек пересечения может получиться не более 2*N^2, каждую точку проверять мы можем за N операций, поэтому на функцию TryPoint уйдет порядка N^3 оперций, что при данных ограничениях, не так много.