# Computational Geometry

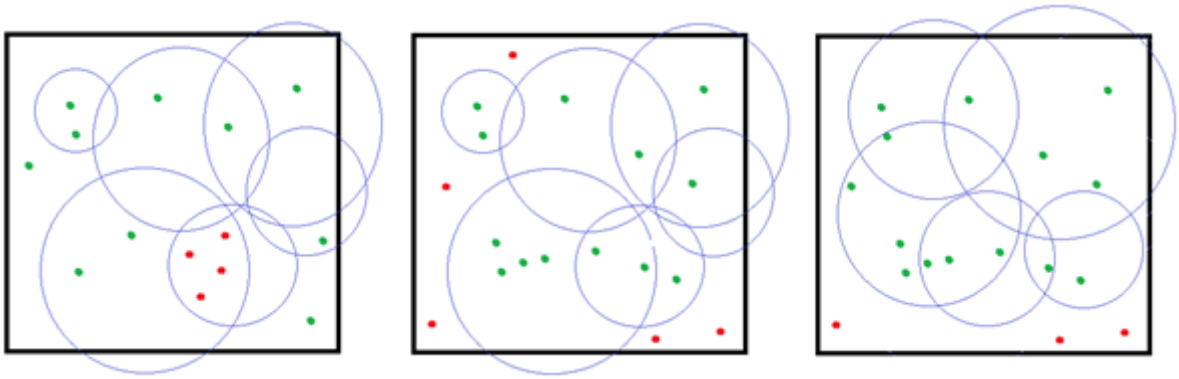
## EX2 – Optimizing Beacons Layout for Localization Applications.

In this assignment we will research the problem of finding the best layout (location and power) of a set of transmitters for positioning applications.

Given a simple polygon (P) representing a region, and n transmitters. We need to assign to each transmitter (t) both: (i) a location with in P, (ii) a positive value representing the transmitter power (the area of the circle centered at t). Such that a localization algorithm based the current visible transmitters will lead to the smallest expected positioning error.

**Definitions:**

- A solution is a set of n circles - all the centers of the circles must be located **within P**.
- Given two solutions (SL1, SL2) for the same polygon P. We can compare between them using the following method: Let T be a set of random points within P, let D1 (D2) be the arrangement of SL1 (SL2), which contains the largest number of points from T. SL1 is better than SL2 (w.r.t. T) iff D1<D2.
- **Goal**: Given a polygon P, an integer n, and a testing set T, find the solution (|SL|=n) which minimizes D(SL,T).



Example: Left: SL1(n=6), T1, D=4, Mid: SL1(n=6), T2,D=5, Right: SL2(n=5), T2, D=3

**Cutting a long story short:**

1. Search the web for two papers which are strongly related to this problem – the closer the better!
2. Solve the 1-dimention problem of n points.
3. Design the best 2-dimantional heuristics / algorithm you can think of.
4. Implement the solution (from #3) to a software package - which support the input / output format as given in the benchmark file. In particular: Input: <n,f_in, f_out>: n number of circles, f_in file name for testing, f_out fine name for output.
   Output: a text file containing n circles in x,y,r format (the same format as the input but with radius).
5. Test your application on the given benchmark – make sure to summarize the experiment results in a table containing both the runtime (T) and quality (the D parameter). E.g.,

| File name | n=4 | n=8 | n=16 | n=32 |
|---|---|---|---|---|
| Input_1 | D,T(4,input_1) | D,T(8,input_1) | D,T(16,input_1) | D,T(32,input_1) |
| … | … | … | … | … |
| Input_k | D,T(4,input_k) | D,T(8,input_k) | D,T(16,input_k) | D,T(32,input_k) |

**General Remarks:**

1. This assignment can be done in groups of up to four students.
2. In this assignment we will only consider the case where P is a the square (0,0), (100,100).
3. In order to simplify the implementation, you can assume the following:
    a. All input / output is in general position.
    b. $n<64$, the size of the solution is limited to 64.