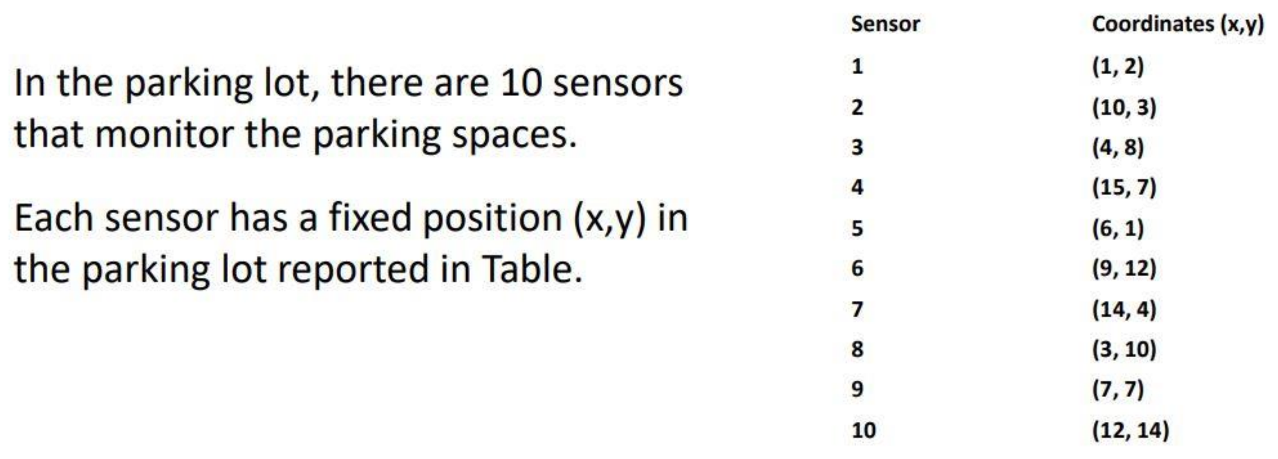
IoT Challenge 1 exercise part:

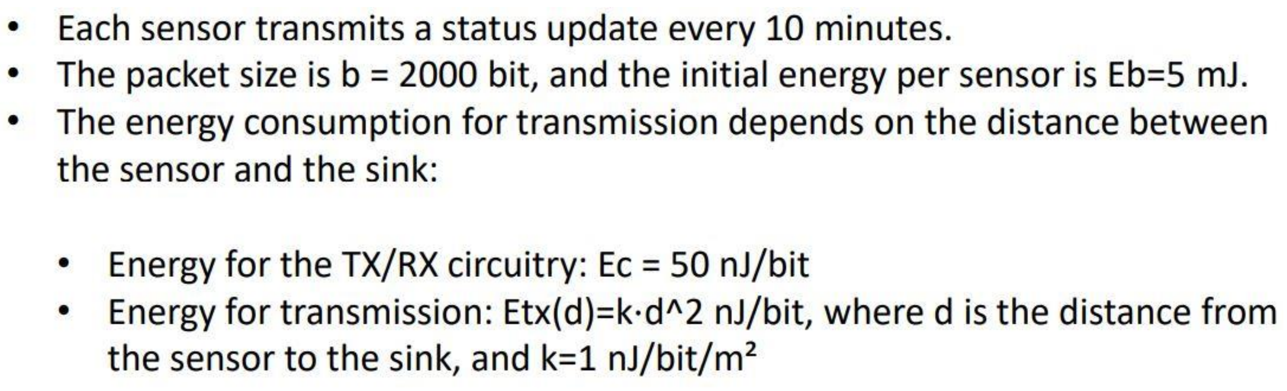
Optimizing Sink Position in a Wireless Sensor Network

Ni Giovanni 10831328 (Leader)
Gu Xinyue 10840236


In the parking lot, there are 10 sensors that monitor the parking spaces.

Each sensor has a fixed position (x,y) in the parking lot reported in Table.

Sensor	**Coordinates (x,y)**
1	**(1, 2)**
2	**(10, 3)**
3	**(4, 8)**
4	**(15, 7)**
5	**(6, 1)**
6	**(9, 12)**
7	**(14, 4)**
8	**(3, 10)**
9	**(7, 7)**
10	**(12, 14)**


- Each sensor transmits a status update every 10 minutes.
- The packet size is b = 2000 bit, and the initial energy per sensor is Eb=5 mJ.
- The energy consumption for transmission depends on the distance between the sensor and the sink:
  - Energy for the TX/RX circuitry: Ec = 50 nJ/bit
  - Energy for transmission: Etx(d)=k·d^2 nJ/bit, where d is the distance from the sensor to the sink, and k=1 nJ/bit/m^2

Question A:

Find the lifetime of the system when the sink is placed at the fixed position (xs,ys)=(20,20). Lifetime is defined as the time until the first sensor's battery dies, based on the energy consumption of the sensors.

Since the sink is placed at a fixed position, we start by calculating the distance of each sensor to the sink. Using the formula for calculating the distance between two points:

$$Distance(d) = \left(\sqrt{}(x1 - x2)^2 + (y1 - y2)^2\right)$$

Sensor	Position	Distance to sink (20,20) [meters]
1	(1,2)	26.17
2	(10,3)	19.72
3	(4,8)	20
4	(15,7)	13.93
5	(6,1)	23.6
6	(9,12)	13.6
7	(14,4)	17.09
8	(3,10)	19.72
9	(7,7)	18.38
10	(12,14)	10

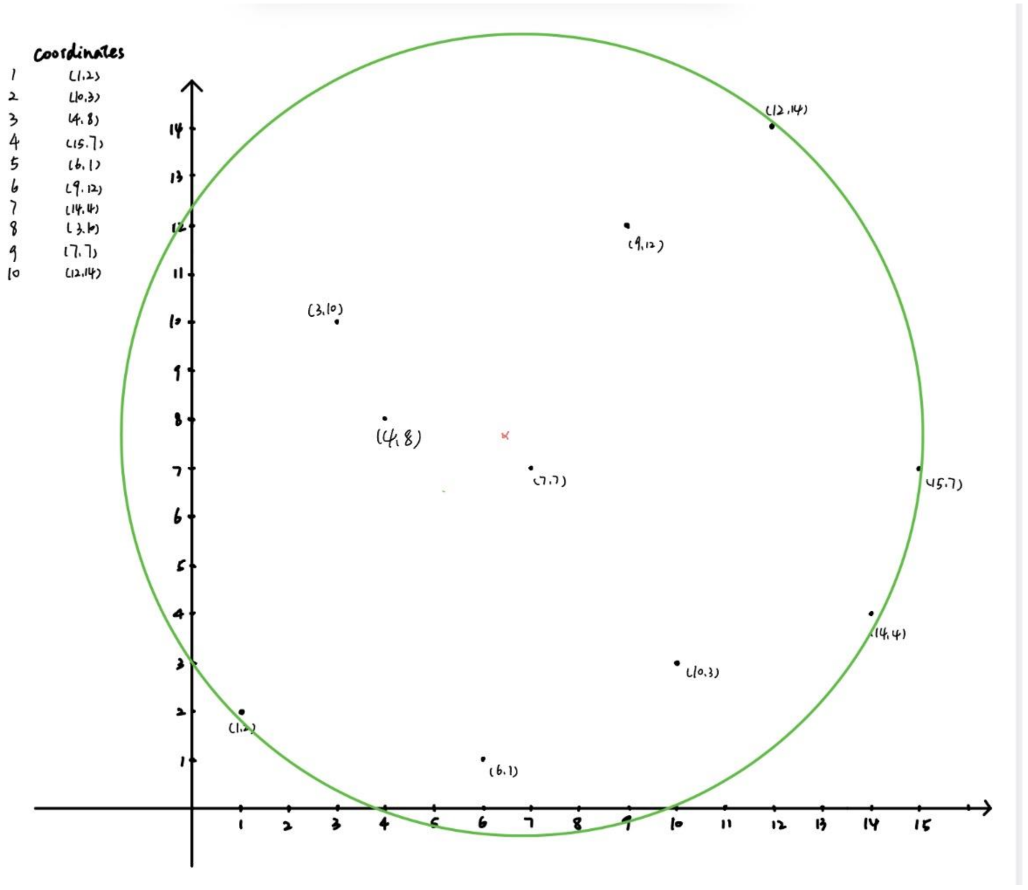
Since the energy for transmission depends on the distance between the sensor and the fixed position of the sink, and we have to find the lifetime of the system, so the time until the first sensor's battery dies based on the energy consumption of the sensors. We could focus only on the farthest sensor to the sink: Sensor 1.

The energy consumed by this node is:

E1 = Ec * b + Etx * b= 50n * 2000 + 1 * (26.17)^2 * 2000n = 1,47mJ

For the lifetime:
Lifetime = Eb/ E1 = 5m/ 1.47m = 3,4cycles.
Since a cycle is 10 minutes, we can complete 3 cycles, so 30 minutes, and at the fourth cycle the sensor consumes all the energy, not finishing the cycle.
Question B:
Find the optimal position of the sink that maximizes the system lifetime. Provide the coordinates (xs,ys) of the sink that minimizes the energy consumption of the worst-case sensor (the sensor that consumes the most energy).

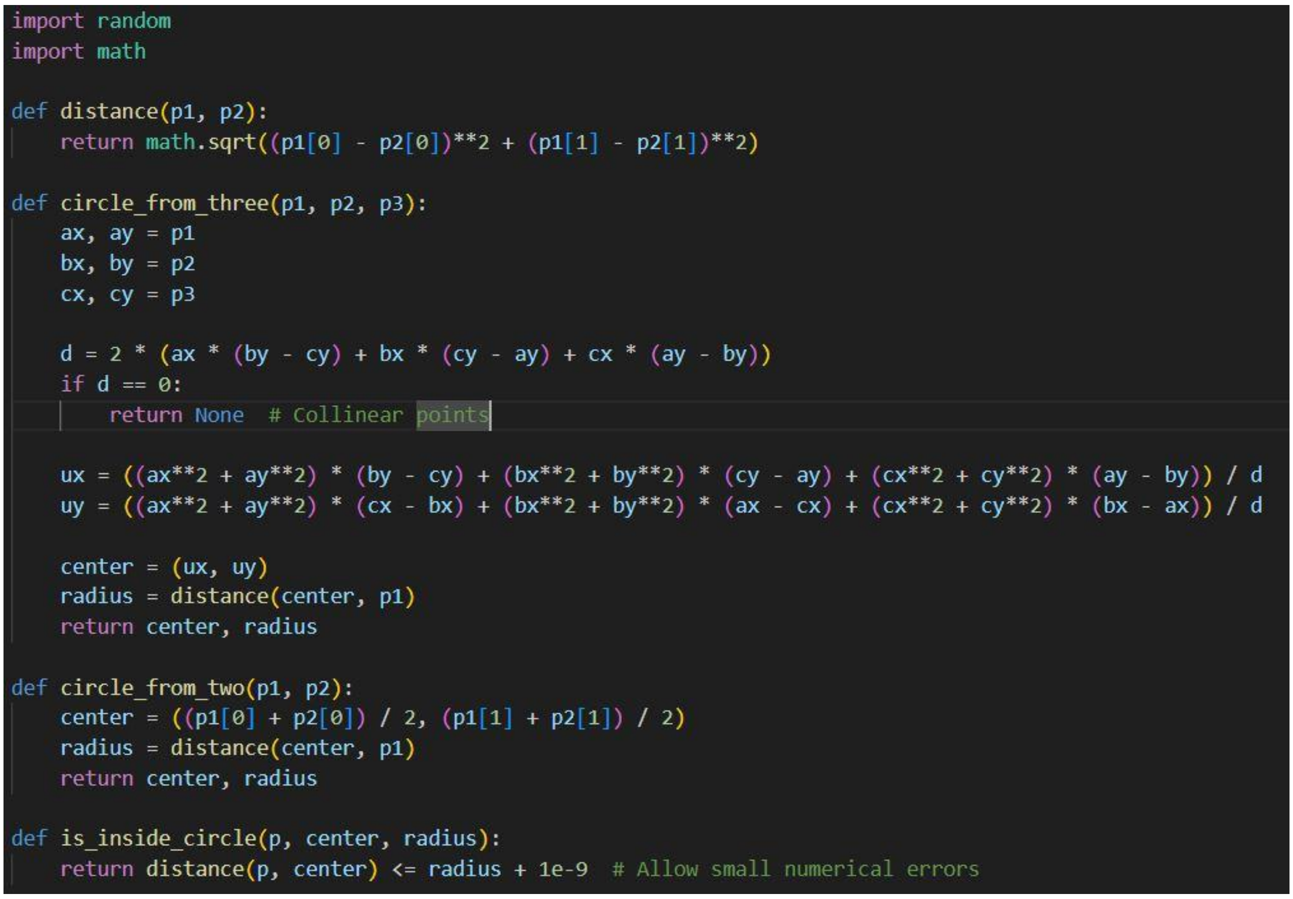
This consist of an optimization problem, our main logic based on "finding the center of the smallest circle that comprehend all the given point"



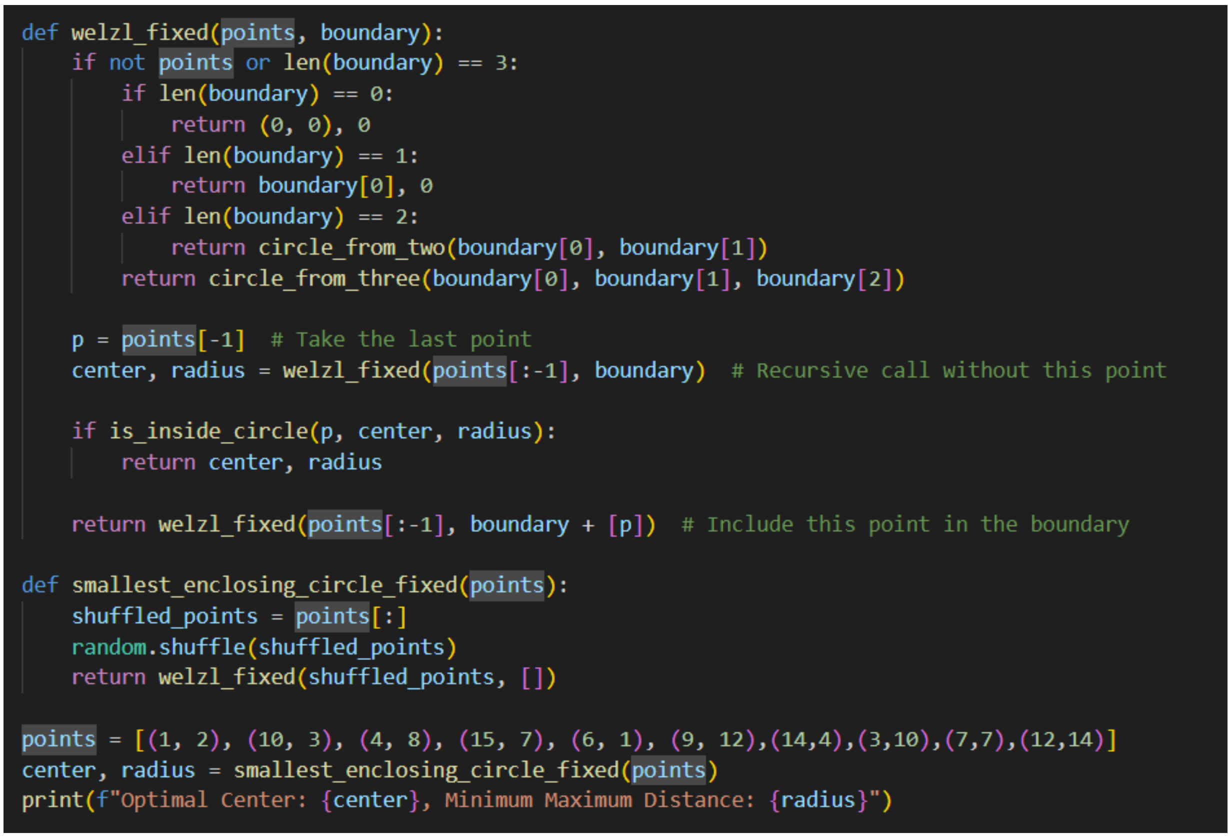
	Coordinates
1	(1,2)
2	(10,3)
3	(4,8)
4	(15,7)
5	(6,1)
6	(9,12)
7	(14,4)
8	(3,10)
9	(7,7)
10	(12,14)

This is our first attempt to draw the circle interested.
Then, we come into the Welzl algorithm that exactly to this.
Welzl's algorithm finds the smallest circle that contains all given points using recursion. It starts with no points in the boundary and adds them one by one. If a point is outside the current circle, the circle is

recalculated with that point as a boundary condition. The result is the smallest enclosing circle, with its center minimizing the maximum distance to any point.
I attached with the report the python code we use to test.

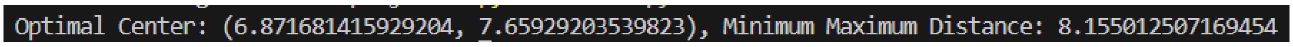


```python
import random
import math

def distance(p1, p2):
    return math.sqrt((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)

def circle_from_three(p1, p2, p3):
    ax, ay = p1
    bx, by = p2
    cx, cy = p3

    d = 2 * (ax * (by - cy) + bx * (cy - ay) + cx * (ay - by))
    if d == 0:
        return None  # Collinear points

    ux = ((ax**2 + ay**2) * (by - cy) + (bx**2 + by**2) * (cy - ay) + (cx**2 + cy**2) * (ay - by)) / d
    uy = ((ax**2 + ay**2) * (cx - bx) + (bx**2 + by**2) * (ax - cx) + (cx**2 + cy**2) * (bx - ax)) / d

    center = (ux, uy)
    radius = distance(center, p1)
    return center, radius

def circle_from_two(p1, p2):
    center = ((p1[0] + p2[0]) / 2, (p1[1] + p2[1]) / 2)
    radius = distance(center, p1)
    return center, radius

def is_inside_circle(p, center, radius):
    return distance(p, center) <= radius + 1e-9  # Allow small numerical errors
```



```python
def welzl_fixed(points, boundary):
    if not points or len(boundary) == 3:
        if len(boundary) == 0:
            return (0, 0), 0
        elif len(boundary) == 1:
            return boundary[0], 0
        elif len(boundary) == 2:
            return circle_from_two(boundary[0], boundary[1])
        return circle_from_three(boundary[0], boundary[1], boundary[2])

    p = points[-1]  # Take the last point
    center, radius = welzl_fixed(points[:-1], boundary)  # Recursive call without this point

    if is_inside_circle(p, center, radius):
        return center, radius

    return welzl_fixed(points[:-1], boundary + [p])  # Include this point in the boundary

def smallest_enclosing_circle_fixed(points):
    shuffled_points = points[:]
    random.shuffle(shuffled_points)
    return welzl_fixed(shuffled_points, [])

points = [(1, 2), (10, 3), (4, 8), (15, 7), (6, 1), (9, 12),(14,4),(3,10),(7,7),(12,14)]
center, radius = smallest_enclosing_circle_fixed(points)
print(f"Optimal Center: {center}, Minimum Maximum Distance: {radius}")
```

Output:



```
Optimal Center: (6.871681415929204, 7.65929203539823), Minimum Maximum Distance: 8.155012507169454
```

The optimal position of the sink that maximizes the system lifetime is: (6.87, 7.66).

Comment:
We notice a very interesting thing. In our case we define as “optimal position” the coordinate of the sink that minimizes the energy consumption of the worst-case sensor, we obtain (6.87, 7.66). Anyway, considering the energy consumption of the entire system (summing all sensors energy consumption), this is not the position that minimize that, for that situation the “centroid” is (8.1, 6.8), instead, the best point where to put the sink, because the centroid is the "balance point" of all the points, which minimizes the sum of squared distances.

Question C:
Discuss the trade-offs involved in choosing a fixed sink position versus dynamically moving the sink. Consider the impact on system lifetime and energy consumption of each sensor.

So, we have two different scenarios:
1. Fixed Sink Position
In the case of a fixed sink, the energy consumption of each sensor is primarily influenced by the distance from the sensor to the sink. The farther a sensor is from the sink, the higher its energy consumption because the signal has to travel a longer distance, and our system lifetime depends on the farthest sensor.
The main advantage of this case is the low complexity because we can avoid the complexity of dynamically controlling the movement of the sink. This can be easier to implement and maintain. But if the sensors are not equally distributed around the sink in terms of distance, which is our case, sensors that are far from the fixed sink will consume more energy to transmit data, draining their batteries faster than those closer to the sink. So, some sensors will run out of battery much sooner than others, resulting in the system lifetime being dictated by the first sensor that dies. And this is not the best way.
Anyway, this solution is optimal when we can find a position where sensors are distributed uniformly around the fixed sink (possibly with low distance).

2. Dynamically Moving Sink
If the sink moves dynamically, it can adapt to the positions of the sensors, potentially reducing the transmission distances and, consequently, the energy consumed by each sensor. The sink could move towards the sensor that is using the most energy, balancing the energy consumption for all sensors, and consequently maximizing the system lifetime. So, for example, every round the sink moves to the sensor that the last round was the farthest, because it was the one which consumed more energy at the last round. It is ideal if sensors send information not together, but one after the other, in this way the sink could be moved every time near to the sensor which is going to send information, but this is just an ideal case.
But we have also to consider some disadvantages. First, moving the sink introduces a more complex system, as you need algorithms to control the movement of the sink, ensure that it doesn’t move too quickly or too slowly, and prevent it from interfering with sensor data communication. Secondly, it’s right that we are minimizing energy consumption for the system, but we should consider also the energy consumption of the sink, the sink will consume energy to move, and this might be substantial, especially if it needs to cover large distances or the energy needed for continuously power on and power off the sink.

And finally, as the sink moves, there could be moments when a sensor is not in the range of the sink, which could cause communication delays or interruptions, leading to potential data loss or the need for retransmission.

To sum up:
Fixed Sink could be simpler and easier to implement but might result in an inefficient use of sensor energy, potentially causing some sensors to die sooner than others and reducing the system’s overall lifetime.
Dynamically Moving Sink can optimize energy usage by reducing the transmission distance, leading to more balanced battery consumption across all sensors extending the system’s lifetime. However, it introduces complexity, energy consumption for the sink's movement, and potential communication delays.
But in our opinion, in this case maximizing system lifetime is the primary goal, so a dynamically moving sink is likely to be the better choice, especially here, where we cannot find a central point for the sink respecting every sensor.