Problem 1. Prove that our 2D MEB problem (i.e., the "no point known" variant) runs in $O(n)$ expected time.

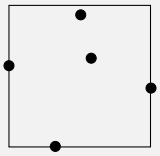
Proof. Fix a subset of the n points $H \subseteq \{p_1, p_2, \ldots, p_n\}$ and $\text{card}(H) = i$. We consider when removing one point in H will cause the minimum enclosing ball to shrink. Suppose there are more than 3 points on the circle, then it removing one point from the circle will not cause it to shrink, since the remaining points uniquely define a circle. Therefore, if the MEB shrinks, then the point being removed can only come from two or three points. The probability is thus $2/i$, $3/i$, resp.

Therefore, the total running time is

$$O(n) + \sum_{i=3}^{n} \mathbb{E}[\mathcal{X}_i] \cdot O(i) \leq O(n) + \sum_{i=3}^{n} \frac{3}{i} \cdot i = O(n),$$

where $\mathbb{E}[\mathcal{X}_\sqsupset]$ is the expected number of changes during the construction. □

Problem 2. Let P be a set of n points in $\mathbb{R}^2$. Find an axis-parallel square of the smallest size to cover the whole P. The figure below shows such a square on an example of $n = 5$.



- Show that this problem can be cast as a linear programming problem with a constant dimensionality.
- Give a deterministic algorithm to solve the problem in $O(n)$ time.

Solution.

- Below constructed a 5D linear programming problem with $4n + 2$ constraints:

$$\begin{aligned} &\textbf{minimize} && z \\ &\textbf{subject to} && z \geqslant x_M - x_m,\ z \geqslant y_M - y_m, \\ &&& x_m \leqslant x_i \leqslant x_M,\ i = 1,\ 2,\ ...,\ n, \\ &&& y_m \leqslant y_i \leqslant y_M,\ i = 1,\ 2,\ ...,\ n. \end{aligned}$$

- A deterministic algorithm is to calculate x_m, x_M, y_m, y_M directly by iterating through all n points in $O(n)$ time. Then, we can derive z by calculating the larger value between $x_M - x_m$ and $y_M - y_m$. The square constructed has at least two (opposite) sides touching a point in P.

Problem 3. Let P be a set of n points in $\mathbb{R}^2$. Give a deterministic algorithm to find in $O(n)$ time an enclosing circle of P with radius at most $\sqrt{2}r^*$, where r^* is the radius of the MEB of P.

Solution. Choose the minimum circle centered at the center of the square with smallest radius covering the square obtained in Problem 2.

r^* is at least half of the side length of the minimum-covering square. Since the circle has to at least touch the square for the two points on the opposite sides.

Since the square suffices to cover the all points in P, the circumcircle of the square must also be able to cover all points in P, the radius is thus at most half of the square's diagonal.

We can conclude that the enclosing circle of P is at most $\sqrt{2}r^*$, since half the diagonal of the square is $\frac{\sqrt{2}}{2}a$ and half the side-length of the square is $a/2$, $r^* \geq a/2$ and $r \leq \frac{\sqrt{2}}{2}$, giving $\frac{r}{r^*} \leq \sqrt{2}$.

Problem 4. Give an algorithm to solve the 3D MEB problem in $O(n)$ expected time.

Solution. A ball is uniquely determined by 4 points that are not coplanar.

Lemma 1. There is only one ball with the smallest radius covering all points in P.

Lemma 2. The boundary of MEB(P) passes at least two points of P.

Proof. Direct from the lecture notes.

Corollary 1. (Uniqueness of Optimal MEBs) The ball covering a set of points P and having $k \geq 0$ points on its boundary is unique.

Lemma 3. Let B_1 and B_2 be two intersecting balls such that C_1 is no larger than C_2. Denote by V the space inside both balls. Consider an arbitrary point p that is covered by B_2 but not B_1. Then, there exists a ball B that is smaller than B_2, passes p, and covers V.

Proof. The centers of these balls having the circle $C = \partial B_1 \cap \partial B_2$ on the boundary must now be on the normal line of C and passes through the center of C. We can place a ball B being divided into left and right curving faces by C and let it grow continuously so that the left curving face moves away from C toward left and the right curing face morphs toward C. By symmetry, we can show the similar behavior. If we ensure that B originally from B_2's position moves toward B_1 and stop as soon as the right curving face hits p. We have constructed the ball B as required.

Lemma 4. Define $P_i = \{p_1, p_2, \ldots, p_i\}$ for each $i \in [1, n]$. For any $i \in [2, n]$, define $B_i^* = \text{MEB}(p_1, \{p_1, p_2, \ldots, p_k\})$. If $p_{i+1} \in B_i^*$, then $B_{i+1}^* = B_i^*$. Otherwise, the boundary of B_{i+1}^* must pass p_{i+1}. This is deemed by Lemma 3.

Make edits to TwoPointsFixedMEB($P, \{p_1, p_2\}$), since the ball passing p_1, p_2, p_i is not unique in 3D. Instead, we let $B \leftarrow$ThreePointsFixedMEB($\{p_1, p_2, \ldots, p_i\}, \{p_1, p_2, p_i\}$) defined as follows.

```
1: procedure ThreePointsFixedMEB(P, {p1, p2, p3})
2:     B ←the smallest ball covering p1, p2, p3
3:     RandomPermutation(P[4, ..., n])
4:     for i ← 4 to n do
5:         if p_i ∉ B then
6:             B ←the ball whose boundary passes p1, p2, p3, p_i
7:         end if
8:     end for
9:     return B
10: end procedure
```

Time complexity analysis, assume the general position that no five points are co-sphere, let $\mathcal{W}_i, \mathcal{X}_i, \mathcal{Y}_i, \mathcal{Z}_i$ be the expected time of the iteration for a specific i, in cases of three, two, one and no points fixed, resp.:

ThreePointsFixedMEB($P, \{p_1, p_2, p_3\}$): the points that define the new MEB can only come from at most 1 point, since we are certain that p_1, p_2, p_3 are on the boundary of the MEB, we have $\mathbb{E}[\mathcal{Z}_i] \leq 1/i$.

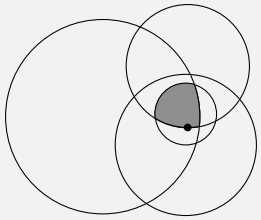
TwoPointsFixedMEB($P, \{p_1, p_2\}$): the points that define the new MEB can only come from at most 2 points, since we are certain that p_1, p_2 are on the boundary of the MEB, we have $\mathbb{E}[\mathcal{Y}_i] \leq 2/i$.

OnePointFixedMEB($P, \{p_1\}$): the points that define the new MEB can only come from at most 3 points, since we are certain that p_1 are on the boundary of the MEB, we have $\mathbb{E}[\mathcal{X}_i] \leq 3/i$.

NoPointsFixedMEB(P): Assume the general position that no five points are co-sphere, then the points that define the new MEB can only come from at most 4 points, we have $\mathbb{E}[\mathcal{W}_i] \leq 4/i$.

Hence, we can conclude that the algorithm runs in $O(n)$ expected time by backward analysis.

Problem 5. Let S be a set of n discs in $\mathbb{R}^2$. Design an $O(n)$ expected time algorithm to find the lowest point that is inside all the discs (or report nothing if there are no such points). Your algorithm may assume that no three circles pass the sample point in $\mathbb{R}^2$.



In the example above, your algorithm should output the black point.

Solution. **Lemma.** The lowest point p inside all discs is either formed by the lowest point of one circle or the lower intersection point of two circles.

Proof. Suppose p lies within the area of the intersection. Then, pulling p downward to the boundary gives a better answer while fulfilling the requirement. Now, p must lie on the boundary of one circle. Then we must be able to drag p down along the circumference until reaching an lower intersection point.

```
1: procedure LowestPointInDiscIntersection(C[1..n])
2:     RandomPermutation(C)
3:     p^(1) ← lowest point of C_1
4:     Pick a random element x from A
5:     for i ← 2 to n do
6:         if p^(i−1) ∈ C_i then
7:             p^(i) ← p^(i−1)
8:         else
9:             if the lowest point of C_i lies in ∩C_i then
10:                p^(i) ← lowest point of C_i
11:            else
12:                p^(i) can only come from the intersection of C_i with some other circles C_j(j < i)
13:                p^(i) ← max{the lower intersection point of C_i ∩ C_j for all j < i}
14:                return none if p^(i) ∉ ∩C_i since the lowest point must be generated by the above
   process
15:            end if
16:        end if
17:    end for
18:    return p^(n)
19: end procedure
```

The point causing the answer to update is drawn randomly with probability $\leq 2/n$, thus it's now apparent to use backward analysis and claim the $O(n)$ expected running time.