P2

Prompt:

Given a set of points $P = \{p_1, ..., p_n\} \in \mathbb{R}^d$, an integer k such that $1 \leq k \leq n$, and the optimum radius O(Δ). Design a polynomial time algorithm that finds at most k balls of radius O(Δ) centered at k points and covering all of the points.

Algorithm:

```
FindSetA(P, O(Δ)):
    Let set S = ∅
    Let radius Δ be 2 * O(Δ)
    While P ≠ ∅ do
        Let point p be arbitrary p ∈ P
        S = S ∪ p
        For each point u ∈ P do
            If distance(p, u) ≤ Δ then
                P = P \ u
            EndIf
        EndFor
    EndWhile
    Return S
```

Claim:

The algorithm terminates in polynomial time.

Proof:

The algorithm has two nested loops. Both loops iterate over the set P that is monotonically decreasing on each iteration. Thus, the algorithm terminates in O(n^2). $\square$

Claim:

Given a set of points P such that $|P| = n$ and the optimum radius O(Δ) to enclose all P in k balls, the algorithm returns a set of points S of size $|S| = h \leq k$ centered at h of the points and covering all of the points.

Proof:

The proof of the claim follows directly. Let O denote the optimum set of points such that all $p \in P$ are within O(Δ) radius from some point $o \in O$ and $|O| = k$. Let S denote the set of points returned by the algorithm at termination. Consider a point $s \in S$ such that $s \in O$. Then, a radius about s of O(Δ) is sufficient to enclose every point $p \in P$ that would also be enclosed by some point

$o \in O$. Obviously, if every point $s \in S$ were a point $o \in O$ we would be done. Now consider the case where point $s \in S$ but $s \notin O$. Let Δ = 2 * O(Δ). Consider the effect doubling O(Δ) has on the points enclosed within a radius Δ from s. Suppose s is O(Δ) away from the nearest $o \in O$. Then, the furthest distance another point q could be from s and still be within O(Δ) of o would be Δ. Therefore, s with radius Δ encloses every point in P that is also enclosed by o with radius O(Δ). It follows then, that at most k points in S with radius Δ will be enough to enclose every point in P just as k points in O with radius O(Δ) were enough to enlose every point in P. □

Prompt:

Assume we no longer know O(Δ) but we know that O(Δ) is in the interval [1, R]. Design an n log(R) polynomial time algorithm to find k balls of radius O(Δ).

Algorithm:

```
FindSetB(P, k):
    Let Δ = 1
    While |S| > k do
        Let S = FindSetA(P, Δ)
        Δ = Δ * 2
    Return S
```

Claim:

The algorithm terminates in n^2 log(R) polynomial time.

Proof:

We know FindSetA terminates in n^2. Since $|S|$ decreases and Δ doubles on every iteration, we know the runtime will be in O(n^2 log(R)). □

Claim:

Given a set P containing n points and an integer k such that $1 \leq k \leq n$, the algorithm returns a set S of at most k points such that all the points in P are within at most c * O(Δ) radius, where c is some constant, from a point in S.

Proof:

Since the algorithm terminates, we know that $|S| \leq k$. We therefore show that Δ is within a constant c of O(Δ). We know that if $\Delta < O(\Delta)$ then $|S| > k$ and we know from part a that if $\Delta > 2 * O(\Delta)$ that $|S| \leq k$. Because we double Δ on each iteration, the largest Δ could be is $\Delta < 4 * O(\Delta)$. However, as shown, no iteration where Δ is larger than that will execute. Thus, $|S| \leq k$ and Δ is bounded by $O(\Delta) <= \Delta < 4 * O(\Delta)$. □