Problem. In Lecture 10, we designed a streaming algorithm for the \( k \)-center clustering problem when points \( p_1, \ldots, p_n \in \{1, \ldots, \Delta\}^d \) are arriving one by one in the stream. For any \( \varepsilon \in (0, 1) \), the algorithm achieves a \((2 + \varepsilon)\)-approximation by storing \( O(k \cdot \log \frac{D}{\varepsilon}) \) points where \( D = \sqrt{d} \cdot \Delta \) is the maximum possible value for the optimum solution. Our goal in this problem is to improve the space complexity of this algorithm at a cost of increasing its approximation ratio by a constant factor.

Design a streaming algorithm for the \( k \)-center clustering problem that achieves an \( O(1) \)-approximation by storing only \( O(k) \) points throughout the stream. Can you reduce the approximation ratio to \((2 + \varepsilon)\)-approximation again by storing only \( O(k/\varepsilon) \) points instead?

Hint: The original approach in the lecture was based on two steps: (i) Designing an \( O(k) \)-space intermediate streaming algorithm that given a parameter \( \tau \in [1, D] \), either outputs a clustering \( C \) with cost at most \( 2 \cdot \tau \), or outputs that the optimal solution has cost more than \( \tau \); (ii) then we did a simple geometric search by running the algorithm above for \( O(\log \frac{D}{\varepsilon}) \) choices of \( \tau \in \{1, (1 + \varepsilon), (1 + \varepsilon)^2, \ldots, D\} \) in parallel.

Modify the second step by performing the geometric search \textit{sequentially} by updating the current guess for \( \tau \) on the fly whenever it is smaller than the optimum value.