Programming Interview for KAIST GSDS (2025 Fall)

Suppose that you are given with an array X of shape (n, d) which represents a set of n points in R^d.

Assume the following python modules are available:

```
import numpy as np
from sklearn.cluster import KMeans
from scipy.spatial import distance
from scipy.optimize import minimize
```

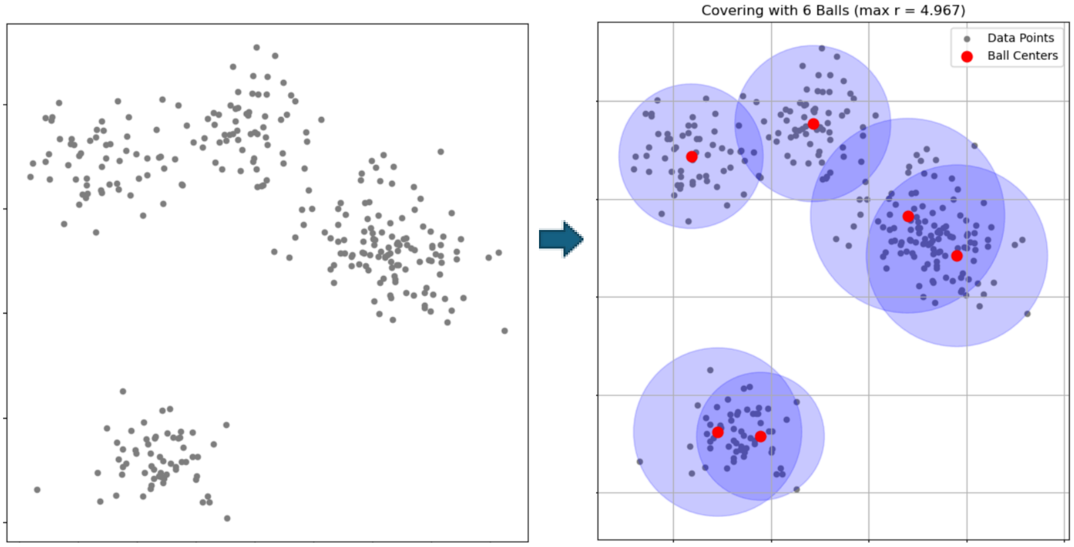
Q1. Fill in the following function (in pseudo-code or python) to find the smallest ball containing all points in X. (Either heuristic approximate solution or exact solution is acceptable.)

```
# Input: X = (n,d) array which represents n points in $R^d$
# output: center ∈ $R^d$, radius ∈ $R^1$
def compute_min_ball(X):
    # An example answer (This is just to exemplify the answer style. The quality
    #  of this answer is not so good. We are expecting better answers from you.)
    # center = X[n//2]
    # radius = |X[0] - center|  # or np.linalg.norm(X[0]-center)



    return center, radius
```

Q2. Assume that `compute_min_ball(X)` function in Q1 is available. We want to find "*cover*" (= a few number of balls with radius < r_max covering all points in X), as exemplified in the figure below. It is desired to cover X with smaller number of balls. Please complete the following `binary_split_cover(X, r_max)` function for finding a cover in divide-and-conquer manner. (Note that you don't have to find the exact optimal cover.)



```
def binary_split_cover(X, r_max):
    centers = []  # list to store ball centers
    radii = []    # list to store ball radii

    def recurse(subset):
        if len(subset) == 0: return








    recu
    retu
```

Q3. Wh

Assume e

number f

necessary.)

O(N log N) : O(N) for each subset, and the recursion depth is O(log N)

T(N) = c_1 N (compute_min_ball) + c_2 N (split) + 2*T(N/2) (for handling two halves)

□ T(N) is O(N log N)

필요한 가정: split에서 거의 N/2로 나눌 수 있다

단, 여기서 N은 정확히는 point 개수가 아니라, X가 차지하는 vol. / ball vol(r_max^d)

□ 모든 가능한 경우를 고려한다면 이 값이 point개수에 비례할 듯..

Q4. Assume that you have an LLM (Large Language Model, such as ChatGPT) which is capable of python coding. Please provide a prompt to the LLM for the code shown in Q2. Since there are many ways of finding a cover, you are supposed to direct the LLM in order to get a code similar to the above. You also need to specify that compute_min_ball(A) function given by yourself should be used. Suppose that you have only 1 chance of asking the LLM, so iterative prompting is not allowed.

(Note that this question is to check whether you can describe problems precisely and concisely. You may compose prompt in Korean.)