

## Practical Python HW

### Description:

You are given a list of tuples of the form ( <float> x, <float> y, <float> r) (let's call these **c-tuples**). Each c-tuple represents a circle on a rectangular coordinate space, with **x** and **y** being coordinates of the center, and **r** being the radius. Assume that each c-tuple has a unique radius.

Let a **cluster** be a group of circles where each circle in the group overlaps with at least one other circle in that group. Two circles with edges that touch tangentially are still considered overlapping. Two circles that aren't overlapping can still be in the same cluster as long as you can traverse from one to another through other overlapping circles.

Write a python script that does the following: For each cluster, the circle with the largest area is kept, and all other circles in that cluster are removed. Return the resulting list of c-tuples. Some examples are shown.

### Requirements:

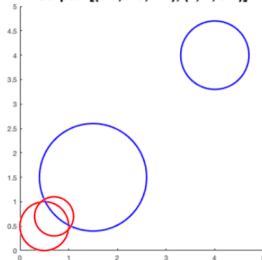
- Provide a range of your code's supported python versions along with justification.
- Only use standard python libraries. You may not use third-party libraries or call any shell/ bash functions.

### Your task:

- Create a python module named **cluster\_utils.py**
- Within this module, provide a top level function named **reduce\_clusters** with the following declaration:

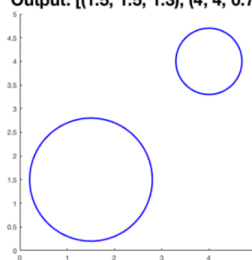
```
def reduce_clusters(ctuple_list):  
    """  
    Takes in an unordered list of ctuples. Identifies clusters of ctuples by overlapping area.  
    Then returns a list of ctuples where each item in the list corresponds to one ctuple per  
    cluster of the highest area.  
  
    The length of the returned list should equal the number of clusters formed from the original  
    tuple list  
  
    :param ctuple_list: A list of ctuple definitions in the form [(x, y, r), (x, y, r), ... ]  
    :return: A list of ctuple definitions in the form [(x, y, r), (x, y, r), ... ]  
    """
```

Input: [(0.5, 0.5, 0.5), (1.5, 1.5, 1.1), (0.7, 0.7, 0.4), (4, 4, 0.7)]  
Output: [(1.5, 1.5, 1.1), (4, 4, 0.7)]



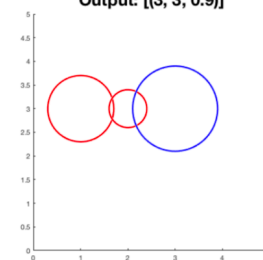
**Figure 1:** The bottom left circles form a cluster. Red circles are removed.

Input: [(1.5, 1.5, 1.3), (4, 4, 0.7)]  
Output: [(1.5, 1.5, 1.3), (4, 4, 0.7)]



**Figure 2:** No clusters are found, and no circles are removed.

Input: [(1, 3, 0.7), (2, 3, 0.4), (3, 3, 0.9)]  
Output: [(3, 3, 0.9)]



**Figure 3:** The three circles form a cluster. Red circles are removed.