Assignment 2

Question 1 (10points). Every Friday night is my movie night. But I have no idea which movie to watch. I just download ratings from IMDB (see the title.ratings.txt file). Please implement a heap sorting algorithm, so that every Friday night I can pick up the unwatched movie with the highest rating. The main function is provided, please complete sort_movies() in another python file. In your code, you are expected to implement heap construction, sorting, output sorted array. You are not allowed to use any off-the-shelf heap packages in Python such as heapq, heappop, heappush, etc.

1. A template has been provided, called heap.py. The main function is in test_heap.py. You're required to practice implementing all essential features of a heap taught in the class, such as maxHeapify, insert, remove, construct a maxheap from an array, etc.
2. You are required to construct a heap in two ways: 1) batch manner, i.e., directly construct it from the whole array; 2) incremental manner, i.e., incrementally adding an item from an array. To distinguish them, you can have two sort_movies functions sort_movies_batch() and sort_movie_incre(). These two functions should both pass the tests.
3. For graduate students, please expand the dataset into 1,000, 10,000, and 100,000. You can try to find this type of movie rating dataset online. If you don't manage to do find one, you can generate some random strings for movie names and random float values for ratings. Test your heap sort in different and large input, measure the runtime and plot Please also compare the performance with some built-in sorting methods in Python(you can make a table or a plot). Please wrap up and submit a small report (in .pdf). No format or template requirement. You need to upload the new data files (name them according to the size: 1,000, 10,000, and 100,000) in .txt format.