After our amazing success we launch a new website. Now everyone can: (i) enter their salary, (ii) query an arbitrary percentile of all salaries. This allows people to better understand the entire distribution of salaries. Which data structure should we use? Provide reasons for your choice, and also argue why the other data structure might be inappropriate.

I would use normal lists and apply qselect() to query the percentile of salaries because its average case complexity is O(n). Building an pecentile-heap would take an average of O(nlogn) because we need O(n) operations to put elements in minh and maxh and need O(logn) operations for each element put to re-heapify. Thus, on average, the qselect() algorithm would work better most of the time.

Generate a short, concise synthesis of the learning (~300 words) from the class session that the student will share with classmates and submit via the form to the instructor.

When we want to find elements at certain locations, there are two algorithms to do so - quickselect and median heap. Median has an advantage over quickselect when we would like to find the median element, and median heap can do so in O(1) (either by finding arithmetics average of minh and max roots or by returning the root of the longer of the two). On the other hand, quick select has a better average case (O(n)) than median heap (O(nlogn) because we would have to rebuild the minh and maxh to suit such percentile) when we would like to query specific percentiles within the data set. To make use of both these pros, we could store data in a median heap and just apply quick select when we want to query something other than the median, then this would treat the median heap as an unsorted list and give average time of O(n).