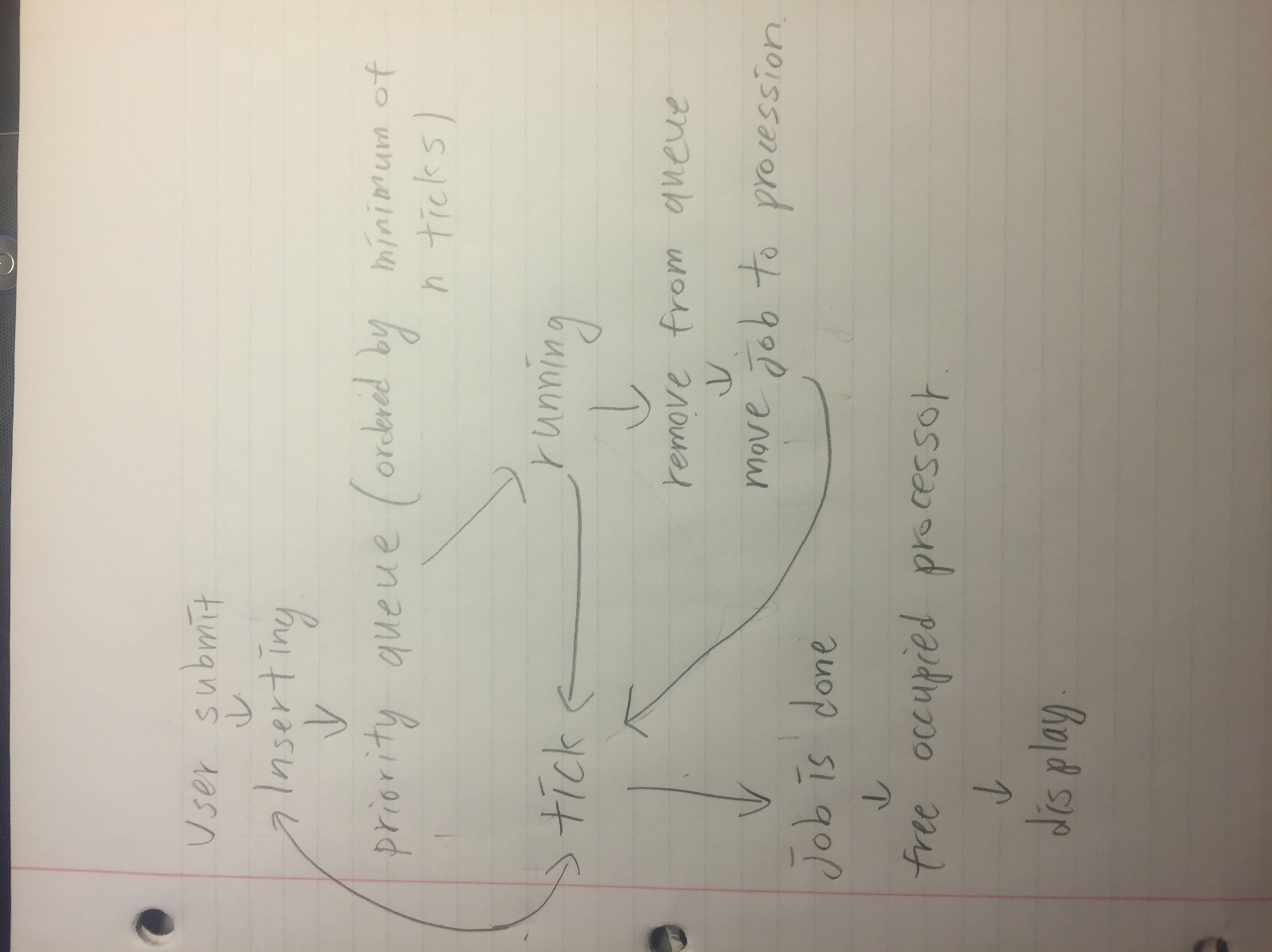
Jeonghyeon Woo

CptS 223

HW 9

Design Document

Wait Queue = Priority Queue

Occupied space list = vector

Worst Cases

InsertJob: O(1) if it includes the worst case of vector function, then O(n)

FindShortest: O(log(n))

DeleteShortest: O(1) if it includes the worst case of vector function, then O(n)

CheckAvailability: O(1) if it include the worst case of FindShortest(), then O(log(n))

RunJob: O(p)

Decrement: O(n of proceeding jobs)

ReleaseProc: O(n of proceeding jobs)

The bottleneck of the queue is that the priority queue is ordered in only one priority. For example, if the first element of queue has big n\_of\_processor, then the whole queue have to stop and wait for the first element’s proceeding. Even if there are free processors, the elements can not be proceed since the elements have bigger n\_of\_ticks.

Last In First Out algorithm is not good when first can not be out but second can be out.