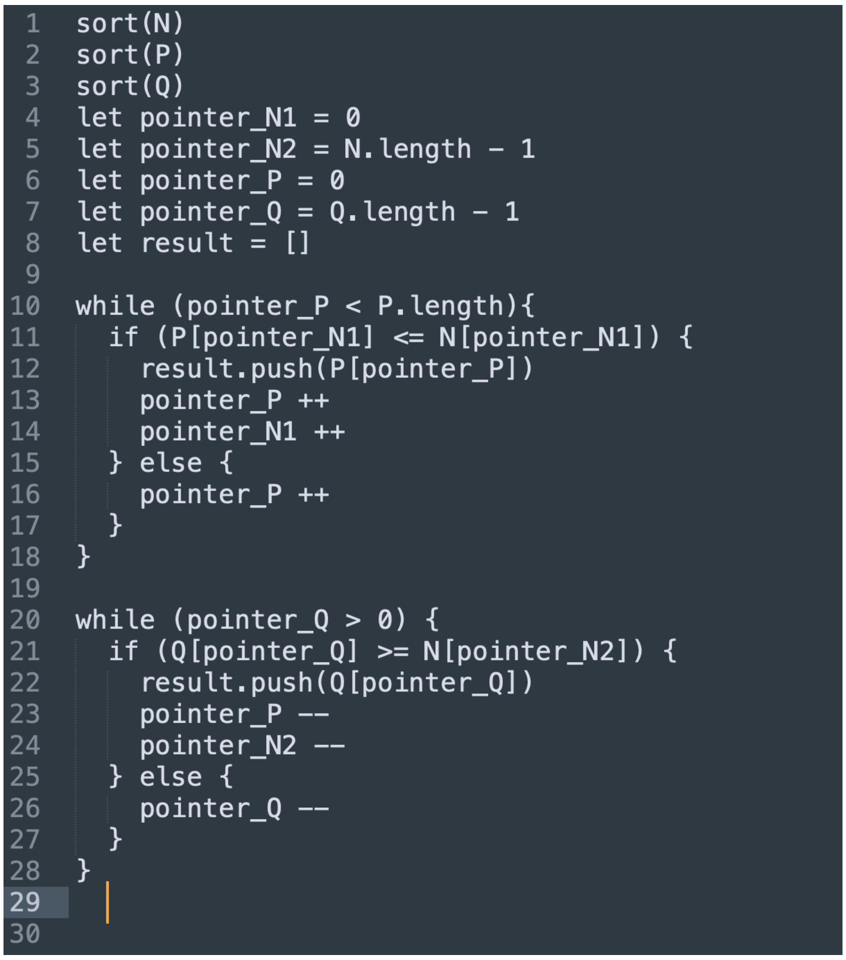
COMP9101　Ass02
Name: Zanning Wang
zID: z5224151

Q2:

In order to assign the largest number of workers to the jobs according to the job requirement. One possible way is that we first assign the lowest x_i to the corresponding P entry-level job, which means, the lowest x_i worker will first get the smallest p_i job among all P entry level job. For senior job, it is similar as the entry level job, we assign the highest x_i worker to the highest q_i job, the sudo-code may show like below:



```
1   sort(N)
2   sort(P)
3   sort(Q)
4   let pointer_N1 = 0
5   let pointer_N2 = N.length - 1
6   let pointer_P = 0
7   let pointer_Q = Q.length - 1
8   let result = []
9
10  while (pointer_P < P.length){
11    if (P[pointer_N1] <= N[pointer_N1]) {
12      result.push(P[pointer_P])
13      pointer_P ++
14      pointer_N1 ++
15    } else {
16      pointer_P ++
17    }
18  }
19
20  while (pointer_Q > 0) {
21    if (Q[pointer_Q] >= N[pointer_N2]) {
22      result.push(Q[pointer_Q])
23      pointer_P --
24      pointer_N2 --
25    } else {
26      pointer_Q --
27    }
28  }
29
30
```

From line1-3, we first sort N,P, Q in ascending order, which the time complexity is O(NlogN+PlogP+QlogQ) ,from line 10 to 18, we maintain two pointer, to find the workers which satisfy the lowest P_i, the time complexity for this step is O(P + N), which is the length of array N and P. similar for the line 20-28, we find the workers which satisfy the requirement Q, the time complexity is also O(N+Q); Compare to the NlogN+PlogP+QlogQ, the time complexity 2N + P +Q should not be consider according to the rules, therefore the total time complexity is O(NlogN+PlogP+QlogQ).