

## 9101 Assignment 2

Haojin Guo

z5216214

### Q2.

2. At a trade school, there are  $N$  workers looking for jobs, each with a skill level  $x_i$ . There are  $P$  entry-level job openings, and the  $i^{th}$  opening only accepts workers with a skill level less than or equal to  $p_i$ . There are also  $Q$  senior job openings, the  $i$  of which requires a skill level of at least  $q_i$ . Each worker can take at most one job, and each job opening only accepts a single worker.

Your task is to determine the largest number of workers you can assign to jobs in time  $O(N \log N + P \log P + Q \log Q)$ .

### Solution,

- 1) Merge Sorting  $N$  and  $P$  in ascending order. (Cost =  $O(N \log N) + O(P \log P)$ )
- 2) Merge Sorting  $Q$  in descending order. (Cost =  $O(Q \log Q)$ )
- 3) Using two pointers to traverse  $N$  and  $P$ .

The worker( $N$ ) in  $N$ , which represent the skill level  $x_i$ .

If  $x_i > \text{skill level } p_i(P)$ , then  $p_i(P)$  ( $p_i(P)++$ ) moves back one position and continues to compare with  $x_i$ .

If  $x_i \leq p_i(P)$ , the number of workers plus one. And then  $p_i(P)$  ( $p_i(P)++$ ) and  $x_i$  ( $x_i++$ ) moves back one position and continues to check.

(The cost is  $N+P$ )

- 4) Using two pointers to traverse  $N$  and  $Q$ .

The worker( $N$ ) in  $N$ , which represent the skill level  $x_i$ .

If  $x_i < \text{skill level } q_i(Q)$ , then  $q_i(Q)$  ( $q_i(Q)++$ ) moves back one position and continues to compare with  $x_i$ .

If  $x_i \geq q_i(Q)$ , the number of workers plus one. And then  $q_i(Q)$  ( $q_i(Q)++$ ) and  $x_i$  ( $x_i++$ ) moves back one position and continues to check.

(The cost is  $N+Q$ )

The total cost is,  $O(N \log N) + O(P \log P) + O(Q \log Q) + 2N + P + Q = O(N \log N) + O(P \log P) + O(Q \log Q)$

```
q2.py
1 def Q2(N_workers, P_level, Q_level):
2     N_workers.sort(reverse=False)
3     P_level.sort(reverse=False)
4
5     find_jobs_p = []
6     for n in N_workers:
7         for p in P_level:
8             if n > p:
9                 continue
10            else:
11                find_jobs_p.append((n, p))
12                N_workers.remove(n)
13                P_level.remove(p)
14                n += 1
15                p += 1
16     Q_level.sort(reverse=True)
17
18     find_jobs_q = []
19     for n in N_workers:
20         for q in Q_level:
21             if n < q:
22                 continue
23            else:
24                find_jobs_q.append((n, q))
25                N_workers.remove(n)
26                Q_level.remove(q)
27                n += 1
28                q += 1
29
30     print("The largest number of workers assigned to jobs is ", int(len(find_jobs_p) + len(find_jobs_q)))
31
32
33 N_workers = [7, 8, 9, 15]
34 P_level = [4, 10]
35 Q_level = [14, 3]
36 Q2(N_workers, P_level, Q_level)
37
```

Q2() > for n in N\_workers > for q in Q\_level > else

Run: q2

/opt/anaconda3/python.app/Contents/MacOS/python /Users/guohaojin/Documents/Python\_Django/q2.py  
The largest number of workers assigned to jobs is 3

Process finished with exit code 0