

Algorithm: Our algorithm is to always allocate easy job for low skill level workers for entry-level jobs, and to always allocate hard job for high skill level workers for senior jobs.

Step 1: Sort workers, entry-level jobs and senior jobs from easy to hard separately according to skill levels. In this way, we get three arrays of workers from  $x_1$  to  $x_N$ , entry-level jobs from  $p_1$  to  $p_P$  and senior jobs from  $q_1$  to  $q_Q$ . For each  $i$  and  $i + 1$  in range, we have  $x_i < x_{i+1}$ ,  $p_i < p_{i+1}$ ,  $q_i < q_{i+1}$ . This step will take time  $O(N \log N + P \log P + Q \log Q)$ .

Step 2: Set a count number,  $a = 0$ . Start with easiest job  $p_1$  and workers with lowest skill level,  $x_1$ , if the worker can do the job, that is  $x_1 \leq p_1$ , allocate the job to the worker and increase  $a$  by 1. Estimate next job,  $p_2$ , with next worker,  $x_2$ . Otherwise, we do not consider  $p_1$  anymore. Instead, estimate next job,  $p_2$ , with the same worker,  $x_1$ .

Step 3: Keep estimating every entry-level job and worker until all entry-level jobs have been estimated or there are no workers without jobs. If no workers left, obviously we find the largest number of workers we can assign to jobs, since every worker has a job. Otherwise, do step 4.

Step 4: Start with hardest job  $q_Q$  and workers with highest skill level,  $x_N$ , if the worker can do the job, that is  $x_N \geq q_Q$ , allocate the job to the worker increase  $a$  by 1. Estimate previous job,  $q_{Q-1}$ , with previous worker,  $x_{N-1}$ . Otherwise, we do not consider  $q_Q$  anymore. Instead, estimate previous job,  $q_{Q-1}$ , with the same worker,  $x_N$ .

Step 5: Keep estimating every senior job and worker until all senior jobs have been estimated or there are workers without jobs. Now  $a$  is the largest number of workers we can assign to jobs.

Time complexity: Note that by this method we only estimate each job once or never, while workers can be estimating many times. The most estimations we can do is when we estimate all entry-level jobs and senior-level jobs with workers, which take time  $O(P + Q)$ . So, the maximum time complexity is  $O(N \log N + P \log P + Q \log Q + P + Q) = O(N \log N + P \log P + Q \log Q)$ .