

Task-01

1. initialize a list to store a pair of values. loop will iterate n times. reading two integers a, b and append it task.
2. `tasks (task[2])` returns second element of task pair where it will be used as a key.
3. Now we will sort it, it checks if the start time $>$ current time or equal. If it is it won't overlap. It will append complete.

Task-02

1. initialize a list `tasks`. The function `ta` returns the second element `task[1]`.
2. we `ta` function as the sorting key. This will ensure tasks are sorted by their end of the time in ascending order.
3. We check if `people[i]` is \leq start time of the current task `task[0]`. If it is met means the person is available. and `complete` is incremented.

Task-03

1. initialize a list queries. The code enters loop that iterates num_queries times. It reads two integers.
2. initialize a dictionary. dictionary is set up with keys representing each person and their values being sets containing only that person.
3. if we find person a and person b not same circle is not equal. it means two needs to be merged as set. to do this set people b is added to a and b is deleted from dictionary. the size of circle-a is added to the list.