//tasks – array of tasks  
Arrays.sort(tasks); //sorting by decreasing price  
TreeSet<Integer> time = new TreeSet<Integer>();  
for (int i = 1; i <= n; ++i) {  
  time.add(i);  
}  
int ans = 0;  
for (int i = 0; i < n; ++i) {  
  Integer tmp = time.floor(tasks[i].time);  
  if (tmp == null) { // If no free space in schedule then place it in the end   
  time.remove(time.last());  
  } else { //else task can be done, add in the schedule  
    time.remove(tmp);  
    ans += tasks[i].cost;  
  }  
}