

4. Implement the optimal algorithm for Weighted Interval Scheduling (for a definition of the problem, see the slides on Canvas) in either C, C++, C#, Java, Python, or Rust. Be efficient and implement it in $O(n^2)$ time, where n is the number of jobs. We saw this problem previously in HW3 Q2a, where we saw that there was no optimal greedy heuristic.

The input will start with a positive integer, giving the number of instances that follow. For each instance, there will be a positive integer, giving the number of jobs. For each job, there will be a trio of positive integers i , j and k , where $i < j$, and i is the start time, j is the end time, and k is the weight.

A sample input is the following:

```
2
1
1 4 5
3
1 2 1
3 4 2
2 6 4
```

The sample input has two instances. The first instance has one job to schedule with a start time of 1, an end time of 4, and a weight of 5. The second instance has 3 jobs.

The objective of the problem is to determine a schedule of non-overlapping intervals with maximum weight and to return this maximum weight. For each instance, your program should output the total weight of the intervals scheduled on a separate line. Each output line should be terminated by exactly one newline. The correct output to the sample input would be:

```
5
5
```

or, written with more explicit whitespace,

```
"5\n5\n"
```

Notes:

- Endpoints are exclusive, so it is okay to include a job ending at time t and a job starting at time t in the same schedule.
- In the third set of tests, some outputs will cause overflow on 32-bit signed integers.