Parth Ray

Weighted Interval Scheduling

a. Let ValWIS(n) represent the maximum total value of a Weighted Interval Scheduling problem of the given size n.
b. ValWIS(n) = max(ValWIS(n - 1), ValWIS(c(n)) + w_n) where c(n) is the job farthest to the right that is compatible with n and where w_n is the weight of n. Base case would be ValWIS(0) = 0.
c. We just need a table of size n + 1 to store each of the jobs' ValWIS plus the base case. Each entry will represent the maximum total value of a weighted interval scheduling problem up to and including that job.
d. FillTable(jobs[1…n]) //jobs sorted in order of increasing finish time

```
ValWIS[0] = 0;

For j = 1 to n

        ValWIS[j] = max(ValWIS(j - 1), ValWIS(c(j)) + wj);

Return ValWIS;
```

e. TraceBack(ValWIS)

```
Trace[];

i = jobs.length;

while (i > 0)

        if (ValWIS(i) == ValWIS(i - 1)) then

                i--;

        else

                trace.prepend(i);

                i = idx(c(i)); //set i to index of c(i), the job farthest right that is

                                compatible with ith job

return trace;
```

f. The complexity of filling in the table is O(n) since we only have to go through the table once.