Solution 1:

For every employee **b** choose **the single cheapest offer that makes someone supervise b**.

If an employee receives no offer, she must be the root.

If more than one employee has no offers, a rooted tree cannot be built.

Algorithm (O(m))

Step	Action
1	Read `n, m`, arrays `qual`, applications.
2	For each triple `(a, b, c)` with `qual[a] > qual[b]` keep `best[b] = min(best[b], c)`.
3	**Feasibility** Exactly one employee must lack an entry in `best`. Condition `best.size() == n − 1`.
4	**Answer** Sum of all `best[b]`; else −1.

Why it's good

Edges never point upward, so any solution can replace its edge into b with the cheapest one without affecting the rest of the tree; thus picking all local minima is globally minimal.

Corner cases

- `n = 1` (no edges needed) → cost 0.
- Duplicate offers for same `b` → keep the smallest.
- Malformed data with upward edges → ignore (not in judge input, but harmless guard).

This was accepted on the first attempt