Suppose that we have an algorithm DTS that correctly solves the Traveling Salesman Decision Problem (TSDP), i.e., given any directed graph G and integer bound B, DTS returns True if G contains a Ham. cycle with total weight at most B; DTS returns False otherwise. Furthermore, let $t(n,m,B)$ be the worst-case running time of DTS on input graphs with n vertices and m edges, and bounds B.

The following algorithm solves TSOP, the Traveling Salesman Optimization Problem (we argue its correctness and analyse its runtime further below):

```
OTS(G, w):
    # First, find the minimum weight of any Ham. cycle in G.
    N ← Σ_{e∈E : w(e)<0} w(e)    # sum of all negative edge weights
    P ← Σ_{e∈E : w(e)>0} w(e)    # sum of all positive edge weights
    if not DTS(G, P):    # there is no Ham. cycle in G at all
        return NIL
    # Perform binary search on the interval [N, P] to find B such that
    # DTS(G, B) = True and DTS(G, B − 1) = False.
    while N < P:    # Loop Inv.: N ⩽ P, DTS(G, N − 1) = False, DTS(G, P) = True
        M ← ⌊(N + P)/2⌋
        if DTS(G, M):    P ← M
        else:            N ← M + 1
    B ← N
    # Next, find the set of edges in a Ham. cycle with total weight B.
    for each e ∈ E:    # Loop Inv.: DTS(G, B) = True
        if DTS(G − e, B):
            G ← G − e
    return E
```

Correctness: The first phase is a standard binary search algorithm. It correctly finds the value of B such that G contains a Ham. cycle with total weight at most B but G contains no Ham. cycle with total weight at most $B-1$.

Given the value of B, the second phase maintains the loop invariant that G contains a Ham. cycle with total weight at most B, so the set of edges returned at the end is guaranteed to contain such a Ham. cycle. Moreover, every edge that is not required for this Ham. cycle will be removed from G, because the loop examines every edge and removes each one that is not necessary.

Hence, the algorithm returns a Ham. cycle in G with the minimum total weight—in other words, it correctly solves TSOP.

Runtime: The first phase iterates $\log_2 W$ times (where $W = \sum_{e\in E} |w(e)|$), making one call to DTS at each iteration, for a total time of $\mathcal{O}\big(\log_2 W \cdot t(n,m,W)\big)$.

The second phase makes one call to DTS and constructs a new graph $G-e$, for each edge $e \in E$. It takes time $\mathcal{O}\big(m \cdot t(n,m,B) + m\cdot(n+m)\big)$, where B is the bound found during the first phase.

Therefore, the total time is $\mathcal{O}\big(\log_2 W \cdot t(n,m,W) + m \cdot t(n,m,B) + m\cdot(n+m)\big)$, which is only a polynomial factor larger than $t(n,m,W)$.