(The “electric car” problem) Given is a directed graph, each arc \((v, w)\) of which has a cost that can be positive, zero, or negative. Also given are two vertices, a source vertex \(s\) and a destination vertex \(t\). Finally, a non-negative battery capacity \(B\) is given. Devise the most efficient algorithm you can to answer the following question: given an electric car starting at vertex \(s\) with a fully charged battery (the battery charge \(b\) is initially \(B\)), can the car be driven to vertex \(t\)? The battery charge \(b\) changes as follows: If the car is at vertex \(v\), it can move from \(v\) to \(w\) along arc \((v, w)\) provided that \(b \geq c(v, w)\). On arrival at \(w\), the battery charge is \(\min\{B, b - c(v, w)\}\). That is, if \(c(v, w) = 0\), the car can move from \(v\) to \(w\) without losing (or gaining) any battery charge. If \(b \geq c(v, w) > 0\), the car can move from \(v\) to \(w\) but its charge drops to \(b - c(v, w)\). If \(c(v, w) < 0\) and \(b - c(v, w) \leq B\), the car can move from \(v\) to \(w\) while increasing its charge to \(b - c(v, w)\). Finally, if \(c(v, w) < 0\) and \(b - c(v, w) > B\), the car can move from \(v\) to \(w\) while increasing its charge, but only to the maximum, \(B\).

As mentioned in class, one can assume that \(B \geq c(v, w)\) for every arc \((v, w)\): any arc whose cost exceeds \(B\) cannot be traversed even with a full battery, and hence all such arcs can be deleted without changing the problem.

For A-level credit, produce an algorithm whose worst-case running time is at most a fixed-degree polynomial in \(n\), the number of vertices, and \(m\), the number of arcs, even if \(B\) and the arc capacities are arbitrary real numbers (assuming addition of two such numbers takes constant time), or provide a reason why such an algorithm might not exist. For B-level credit, produce an algorithm whose worst-case running time is polynomial in \(n, m,\) and \(B\), assuming \(B\) and all the arc costs are integers, or provide a reason why such an algorithm might not exist.