```cpp
const int MAXN = 10000;
int memo[MAXN];

// Definisce la struttura arco_t, che contiene
// i nodi di partenza e di arrivo di un arco
struct arco_t {
    int u, v;

    arco_t(int u, int v): u(u), v(v) {}
};

int accelera(int N, int M, int da[], int a[], int V[]) {
    map <int, vector<arco_t>, greater<int> > archi;

    // Inserisce gli archi nella map, raggruppandoli in base alla lunghezza
    for (int i = 0; i < M; i++)
        archi[V[i]].emplace_back(da[i], a[i]);

    unordered_map <int, int> aggiornamento;
    for (const auto& gruppi: archi) {           // Per ogni gruppo
        for (const auto& arco: gruppi.second) { // Per ogni arco
            int u = arco.u;
            int v = arco.v;

            // aggiornamento[u] contiene la lunghezza
            // del cammino massimo che parte da u
            if (!aggiornamento.count(u))
                aggiornamento[u] = memo[u];
            aggiornamento[u] = max(aggiornamento[u], 1 + memo[v]);
        }

        // Trasferisce i valori a memo
        for (const auto& it: aggiornamento)
            memo[it.first] = it.second;
    }

    // La soluzione è il massimo dei valori presenti in memo
    return *max_element(memo, memo + N);
}
```