

```

branch&bound(Item[] S, int n, int i, int minCost, int minSol)
    SET C ← scelte(S, n, i, ...) // determina l'insieme in funzione delle scelte precedenti
    // esamino ogni scelta
    foreach c ∈ C do
        S[i] ← c

        // calcolo il lower bound
        int lb ← lb(S, i) // calcolato in base alle scelte fatte fin'ora
        if lb < minCost then
            // il limite inferiore non eccede il costo minimo
            if i < n then
                // sono arrivato "in fondo"

                // faccio ricorsivamente le scelte successive
                branch&bound(S, n, i + 1, minCost, minSol)
            else if c(S, i) < minCost then
                // la soluzione trovata è migliore del minimo parziale
                minSol ← S // aggiorno la soluzione minima parziale
                minCost ← c(S, i) // aggiorno il costo minimo parziale

```