

```
bbTsp([ S, int cost, SET R, int n, int i)
```

```
SET choices  $\leftarrow$  copy(R)
```

```
foreach c  $\in$  choices do
```

```
    S[i]  $\leftarrow$  c // memorizza la scelta
```

```
    // MECCANISMO PER GENERARE LE PERMUTAZIONI
```

```
    R.remove(c) // rimuovi la scelta nell'insieme delle scelte possibili
```

```
    // MECCANISMO DI VALUTAZIONE
```

```
    if i < n then
```

```
        { calcola out, back, transfer[h] per ogni h  $\in$  R  $\Leftrightarrow$  h  $\notin$  S[1...i] }
```

```
        int lb  $\leftarrow$   $\left\lceil \frac{out + \sum_{h \notin S} transfer[h] + back}{2} \right\rceil$ 
```

```
        if lb < minCost then
```

```
            bbTsp(S, cost + d[S[i - 1]][S[i]], R, i + 1)
```

```
    else
```

```
        lb  $\leftarrow$  cost + d[S[i]][S[1]] // aggiorna il limite inferiore
```

```
        if lb < minCost then // se abbiamo trovato una soluzione migliore
```

```
            minSol  $\leftarrow$  copy(S) // aggiorna la soluzione ottimale
```

```
            minCost  $\leftarrow$  lb // aggiorna il costo minimo
```

```
    R.insert(c) // reinserisco la scelta nell'insieme delle scelte possibili
```