

## Allgemeiner Backtracking Algorithmus

```
boolean FindeLoesung(int index, Lsg loesung, ...) {
    // index ist die aktuelle Schrittzahl
    // Teillösungen loesung werden als Referenz übergeben.

    1. Solange es noch neue Teil-Lösungsschritte gibt:
        a) Wähle einen neuen Teil-Lösungsschritt schritt; // Heuristik
        b) Falls schritt gültig ist:
            I) Erweitere loesung um schritt;
            II) Falls loesung vollständig ist, return true, sonst:
                if (FindeLoesung(index+1,loesung)) { // rekursiv
                    return true; // Lösung gefunden
                } else { // Wir sind in einer Sackgasse
                    Mache schritt rückgängig; // Backtracking
                }

    2. Gibt es keine neuen Teil-Lösungsschritte mehr, so: return false
}
```

## Implementierung in Java-Pseudocode

```
boolean FindeLoesung(int index, Lsg loesung, ...) {
    // index = Schrittzahl
    // loesung = Referenz auf Teillösung

    while (es gibt noch neue Teil-Lösungsschritte) {
        Wähle einen neuen Teil-Lösungsschritt schritt; // Heuristik
        if (schritt ist gültig) {
            Erweitere loesung um schritt;

            if (loesung noch nicht vollständig) {
                // rekursiver Aufruf von FindeLoesung

                if (FindeLoesung(index+1,loesung,...)) {
                    return true; // Lösung gefunden
                } else { // wir sind in einer Sackgasse
                    Mache schritt rückgängig; // Backtracking
                }
            } else return true; // Lösung gefunden -> fertig
        }
    }
    return false;
} // Bei true als Rückgabewert steht die Lösung in loesung
```