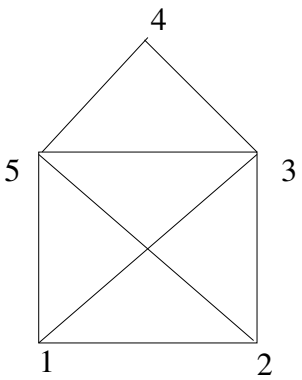
Backtracking

Problemstellung



Hier ist das „Haus des Nikolaus“ mit einer bestimmten Nummerierung der Eckpunkte vorgegeben. Es sollen alle Lösungen zum Zeichnen der Figur in einem Zug gefunden werden. Eine Lösung könnte dann in der Form 123451352 ausgegeben werden. Das Programm soll eine einfache Anpassung an andere Graphen ermöglichen. Der Ausschluss von gespiegelten Lösungen ist nicht gefordert.

Backtracking

Eine Lösung lässt sich nach dem Prinzip *Versuch und Testen* ermitteln. Eine vermutete Teillösung muss wieder verworfen werden, wenn ein Test ihre Ungültigkeit nachgewiesen hat. Man nennt diesen Ansatz deshalb auch *Rückverfolgen* oder *Backtracking*. Mit diesem Ansatz lassen sich eine ganze Reihe von Problemen in der Informatik sehr elegant formulieren und lösen. Hier eine kleine Auswahl (Genaueres dazu später):

- *Acht-Damen-Problem:* Acht Damen sollen so auf ein Schachbrett gestellt werden, dass keine Dame eine andere bedroht.
- *Vier-Farben-Problem:* Eine Landkarte soll mit vier Farben so gefärbt werden, dass benachbarte Länder immer unterschiedliche Farben bekommen.
- *Labyrinth-Problem:* Ein Labyrinth mit Sackgassen und Verzweigungen ist zu durchlaufen, um den Ausgang zu finden.

Konkreter:

1. Man versucht, eine Kante (Verbindungsstrecke) zu zeichnen, wenn sie zulässig ist oder noch nicht gezeichnet wurde.
2. Ist das nicht möglich, muss die zuletzt gezeichnete Kante gelöscht werden.
3. Ist es möglich, dann hat man das Problem um eine Stufe vereinfacht.
4. Hat man durch dieses Verfahren insgesamt 8 Kanten zeichnen können, hat man eine Lösung gefunden. Jetzt löscht man wieder die zuletzt gezeichnete Kante und sucht nach weiteren Lösungen.

Realisierung des Programms

Datenstrukturen

	1	2	3	4	5
1		X	X		X
2	X		X		X
3	X	X		X	X
4			X		X
5	X	X	X	X	

Die Tabelle gibt an, welche Verbindungslinien zulässig sind (durch X markiert). Die erste Zeile bedeutet also, dass von Punkt 1 zu den Punkten 2, 3 und 5 Strecken gezeichnet werden dürfen. Eine solche Tabelle heißt auch *Adjazenzmatrix* (von adjazieren; lat.: anwohnen, anliegen). Eine solche Tabelle lässt sich durch

```
boolean[][] kanteZulaessig;
```

in einem zweidimensionalen Feld speichern. Eine entsprechende Tabelle

```
boolean[][] kanteGezeichnet;
```

erfasst dann die schon gezeichneten Kanten. In einem weiteren eindimensionalen Feld wird jeweils eine Lösung erfasst.

Methoden

Es bieten sich folgende Methoden zur Strukturierung des Programmes an:

1. `void initArrays()`
2. `void zeichneKante(int von, int nach)`
3. `void loescheKante(int von, int nach)`
4. `void loesungAusgeben()`
5. `void versucheKanteZuZeichnen(int start)` : Die rekursive Methode soll vom Punkt `start` weitere Kanten zeichen.
6. Das Hauptprogramm:

```
public static void main(String[] arg)
{ initArrays();
  for (int punktNr=1; punktNr<=maxPunktAnz; punktNr++)
  {  loesungsWeg[0] = punktNr; // Startpunkt eintragen
     versucheKanteZuZeichnen(punktNr);
  } // for punktNr
  Out.println();
  Out.println("Es ergaben sich "+loesungsAnzahl+" Loesungen.");
} // main
```