

Das Fischrätzel

Ein typischer Vertreter beliebter logischer Rätsel, ist die folgende Aufgabe:

Es gibt fünf Häuser mit je einer anderen Farbe.
 In jedem Haus wohnt eine Person einer anderen Nationalität.
 Jeder Hausbewohner bevorzugt ein bestimmtes Getränk, raucht eine bestimmte Zigarettenmarke und hält ein bestimmtes Haustier.
 Keine der 5 Personen trinkt das gleiche Getränk, raucht die gleichen Zigaretten oder hält das gleiche Tier wie einer seiner Nachbarn.

Die Hinweise:

1. Der Brite lebt im roten Haus.
2. Der Schwede hält einen Hund.
3. Der Däne trinkt gerne Tee.
4. Das grüne Haus steht links neben dem weißen Haus.
5. Der Besitzer des grünen Hauses trinkt Kaffee.
6. Die Person, die Pall Mall raucht, hält einen Vogel.
7. Der Mann, der im mittleren Haus wohnt, trinkt Milch.
8. Der Besitzer des gelben Hauses raucht Dunhill.
9. Der Norweger wohnt ganz links im ersten Haus.
10. Der Marlboro-Raucher wohnt neben dem, der eine Katze hält.
11. Der Mann, der ein Pferd hält, wohnt neben dem, der Dunhill raucht.
12. Der Winfield-Raucher trinkt gerne Bier.
13. Der Norweger wohnt neben dem blauen Haus.
14. Der Deutsche raucht Rothmanns.
15. Der Marlboro-Raucher hat einen Nachbarn, der Wasser trinkt.

Die Frage: Wem gehört der Fisch?

Hier gibt es 6 Kategorien: Nation, Haustier, Getränk, Zigarette, Hausnummer, Farbe. In jeder Kategorie gibt es 5 zuzuordnende Begriffe.

Wie viele Möglichkeiten der Zuordnung gibt es: pro Kategorie $5! = 120$ Stück, das ganze in 6 unabhängigen Kategorien. Insgesamt also $120^6 = 2985984000000$, also fast 3 Billionen Möglichkeiten. Hier dürfte stures Durchprobieren aller Möglichkeiten wohl nicht zum Ziel führen.

Erste Reduktion:

Ganz so viele müssen wir zum Glück nicht betrachten. Eine Kategorie, etwa die Nationalitäten, können wir fest in einer beliebigen Reihe vorgeben.

Nehmen wir etwa fix an:

Brite, Schwede, Däne, Norweger, Deutscher = 1,2,3,4,5

Weiters gibt es Regeln, die eine Zuordnung bereits eindeutig vorgeben:

Tip 2 etwa sagt: Der Hund gehört zum Schweden.

Da es für jede Kategorie so eine Information gibt, verbleiben nur mehr $(4!)^5 = 7962624$, also etwa 8 Millionen durchzutestende Möglichkeiten. Das klingt schon besser!

Wie wollen wir die Informationen im Programm darstellen:

Wenn der Hund zum Schweden gehört, können wir eine Variable `hund` mit dem Wert 2 belegen.

Wir bilden also Zugehörigkeiten jedes Objekts zu einer Person. (Wir hätten aber auch die Tiere oder

die Getränke als Basis nehmen können).

Extra Aufmerksamkeit brauchen die Hausnummern. Wohnt der Deutsche in Nummer 3, so könnten wir in analoger Weise zu den anderen Eigenschaften `nummer3 = 5` sagen. Aber dann gestaltet sich das Vergleichen der Nummern (links von bzw. kleiner, oder neben bzw. Nummernunterschied 1) schwierig. Drehen wir besser um und nehmen eine Liste namens `nummer`, mit der wie die Information als `nummer[5]=3` formulieren.

Dann wird Tip 13 einfach zu `abs (nummer[norweger] - nummer[blau]) == 1`

Permutationen:

Um alle möglichen Zuordnungen der Getränke bilden zu können, benötigen wir alle möglichen Anordnungen der auftretenden Worte, das nennt man Permutationen.

Beispiel:

Die Permutationen von abc sind: abc, acb, bac, bca, cab, cba.

Eine Funktion, die alle Permutationen einer beliebigen Liste ausgibt, ist etwa (`teste perm([1,'b','ce'])`)

```
def perm(x):
    if not x: return [x]
    result = []
    for i in range(len(x)):
        rest = x[:i]+x[i+1:]
        for r in perm(rest):
            result.append(x[i:i+1]+r)
    return result
```

Für unsere Logikaufgabe wähle ich aber, da wir nie mehr als 4 Elemente brauchen, eine Version mit maximaler Laufgeschwindigkeit:

```
def perm(x):
    l=len(x)
    if l==1: return x
    elif l==2:
        a,b = x
        return [[a,b],[b,a]]
    elif l==3:
        a,b,c = x
        return [[a,b,c],[a,c,b],[b,a,c],[b,c,a],[c,a,b],[c,b,a]]
    elif l==4:
        a,b,c,d = x
        return [[a,b,c,d],[a,c,b,d],[b,a,c,d],[b,c,a,d],[c,a,b,d],[c,b,a,d],
                [a,b,d,c],[a,c,d,b],[b,a,d,c],[b,c,d,a],[c,a,d,b],[c,b,d,a],
                [a,d,b,c],[a,d,c,b],[b,d,a,c],[b,d,c,a],[c,d,a,b],[c,d,b,a],
                [d,a,b,c],[d,a,c,b],[d,b,a,c],[d,b,c,a],[d,c,a,b],[d,c,b,a]]
```

Hier habe ich einfach alle Möglichkeiten mit der Hand hingeschrieben. Nicht elegant, aber für unsere Zwecke effizient. (Für alle Eventualitäten könntest Du beide Versionen in Dein Programm aufnehmen, die erste Version `perm1` nennen, und in `perm` die Zeile `else: return perm1(x)` anfügen – dann klappt die Sache in jedem Fall. Beim Fischrätzel wäre es aber übertrieben...

Erste Version

```

def solve():

    nummer = [0]*6                                # Hausnummern
    brite,schwede,daene,norweger,deutscher=1,2,3,4,5
    rot=1                                           # Tip 1
    hund=2                                         # Tip 2
    tee=3                                           # Tip 3
    nummer[4]=1                                    # Tip 9
    rothmans=5                                     # Tip 14
    for (pallmall,marlboro,dunhill,winfield) in perm([1,2,3,4]):

        for (vogel,katze,pferd,fisch) in perm([1,3,4,5]):

            for (kaffee,milch,bier,wasser) in perm([1,2,4,5]) :

                for (gruen,weiss,gelb,blau) in perm([2,3,4,5]) :

                    for (nummer[1],nummer[2],nummer[3],nummer[5]) in perm([2,3,4,5]):

                        if nummer[gruen]!=nummer[weiss]-1: continue          # Tip 4
                        if kaffee!=gruen: continue                          # Tip 5
                        if vogel!=pallmall: continue                        # Tip 6
                        if nummer[milch]!=3: continue                       # Tip 7
                        if dunhill!=gelb: continue                          # Tip 8
                        if abs(nummer[marlboro]-nummer[katze])!=1: continue # Tip 10
                        if abs(nummer[dunhill]-nummer[pferd])!=1: continue # Tip 11
                        if winfield!=bier: continue                         # Tip 12
                        if abs(nummer[norweger]-nummer[blau])!=1: continue # Tip 13
                        if abs(nummer[marlboro]-nummer[wasser])!=1: continue # Tip 15

                        print
                        print "brit,schw,dae,nor,deu
",brite,schwede,daene,norweger,deutscher
                        print "hund,vogel,katze,pferd,fisch
",hund,vogel,katze,pferd,fisch
                        print "tee,kaffee,milch,bier,wasser
",tee,kaffee,milch,bier,wasser
                        print "pallmall,marlboro,dunhill,winfield,rothmans
",pallmall,marlboro,dunhill,winfield,rothmans
                        print "rot,gruen,weiss,gelb,blau
",rot,gruen,weiss,gelb,blau
                        print "nummern                                ",nummer[1:]

```

Toll! Wir bekommen eine Lösung. Doch die Laufzeit ist lang. Auf meinem (alten) Rechner dauert es etwa 7 Minuten.

Was bedeutet die Ausgabe:

```

brit,schw,dae,nor,deu          1 2 3 4 5
hund,vogel,katze,pferd,fisch
tee,kaffee,milch,bier,wasser  3 5 1 2 4
pallmall,marlboro,dunhill,winfield,rothmans  1 3 4 2 5
rot,gruen,weiss,gelb,blau     1 5 2 4 3
nummern                        3 5 2 1 4

```

Wasser hat Nummer 4, das ist der Norweger, der im gelben Haus mit der Nummer 1 (Vorsicht – umgekehrte Zuordnung) wohnt und Dunhill raucht. Sein Haustier verrate ich hier nicht.

Verbesserung:

Ein wenig Überlegung bringt viel! Es gibt doch Hinweise, die wir schon vor der innersten Schleife testen können. Etwa ist der Tip 6 prüfbar, sobald die Tiere und die Zigaretten in der Schleife auftauchen. Ziehen wir diese Abfragen nach außen, sparen wir Zeit!

```
def solve():
    nummer = [0]*6                                # Hausnummern
    brite,schwede,daene,norweger,deutscher=1,2,3,4,5
    rot=1                                          # Tip 1
    hund=2                                        # Tip 2
    tee=3                                         # Tip 3
    nummer[4]=1                                   # Tip 9
    rothmans=5                                    # Tip 14
    for (pallmall,marlboro,dunhill,winfield) in perm([1,2,3,4]):
        for (vogel,katze,pferd,fisch) in perm([1,3,4,5]):
            if vogel!=pallmall: continue         # Tip 6
        for (kaffee,milch,bier,wasser) in perm([1,2,4,5]) :
            if winfield!=bier: continue         # Tip 12
        for (gruen,weiss,gelb,blau) in perm([2,3,4,5]) :
            if kaffee!=gruen: continue         # Tip 5
            if dunhill!=gelb: continue         # Tip 8
        for (nummer[1],nummer[2],nummer[3],nummer[5]) in perm([2,3,4,5]):
            if nummer[gruen]!=nummer[weiss]-1: continue # Tip 4
            if nummer[milch]!=3: continue     # Tip 7
            if abs(nummer[marlboro]-nummer[katze])!=1: continue # Tip 10
            if abs(nummer[dunhill]-nummer[pferd])!=1: continue # Tip 11
            if abs(nummer[norweger]-nummer[blau])!=1: continue # Tip 13
            if abs(nummer[marlboro]-nummer[wasser])!=1: continue # Tip 15

        print
        print "brit,schw,dae,nor,deu
",brite,schwede,daene,norweger,deutscher
        print "hund,vogel,katze,pferd,fisch
",hund,vogel,katze,pferd,fisch
        print "tee,kaffee,milch,bier,wasser
",tee,kaffee,milch,bier,wasser
        print "pallmall,marlboro,dunhill,winfield,rothmans
",pallmall,marlboro,dunhill,winfield,rothmans
        print "rot,gruen,weiss,gelb,blau
",rot,gruen,weiss,gelb,blau
        print "nummern
",nummer[1],nummer[2],nummer[3],nummer[4],nummer[5]
```

Nicht schlecht! Jetzt kommt das Ergebnis erfreulich bald!

Spitzentempo:

Stecken wir noch ein wenig mehr Hirn ins Programm: Versuchen wir alle Informationen möglichst früh zu berücksichtigen!

Kennen wir etwa bereits 2 Getränke, müssen doch nur mehr die 3 übrigen in der Permutationsliste auftauchen. Wir entfernen also alle bereits bekannten Objekte aus den Kategorien. Allerdings müssen wir dann auch selbst dafür sorgen, dass die Regeln eingehalten werden: Man kann nur Objekte aus der Liste entfernen, wenn sie drin sind – andernfalls hätten wir eine Doppelzuordnung die nicht sein kann, also eine Unmöglichkeit. Auch dürfen nicht 2 Objekte der gleichen Kategorie einer einzigen Person zugeordnet werden.

Das klingt komplizierter als es ist:

```
def solve():
    nummer = [0]*6                                # Hausnummern
    brite,schwede,daene,norweger,deutscher=1,2,3,4,5
    rot=1                                          # Tip 1
    hund=2                                        # Tip 2
    tee=3                                         # Tip 3
    nummer[4]=1                                   # Tip 9
    rothmans=5                                    # Tip 14
    for (pallmall,marlboro,dunhill,winfield) in perm([1,2,3,4]):
        vogel=pallmall                           # Tip 6
        gelb=dunhill                             # Tip 8
        bier=winfield                             # Tip 12
        getraenke = [1,2,4,5]                    # ohne tee=3
        if not bier in getraenke: continue        # waere bier=3
        getraenke.remove(bier)
        for (kaffee,milch,wasser) in perm(getraenke):
            gruen=kaffee                          # Tip 5
            if gruen==gelb: continue
            farben=[2,3,4,5]
            if not gelb in farben: continue
            if not gruen in farben: continue
            farben.remove(gelb)
            farben.remove(gruen)
            for (weiss,blau) in perm(farben):
                tiere = [1,3,4,5]
                if vogel==hund: continue
                if not vogel in tiere: continue
                tiere.remove(vogel)
                for (katze,pferd,fisch) in perm(tiere):
                    for nummer[1],nummer[2],nummer[3],nummer[5] in perm([2,3,4,5]):
                        if nummer[milch]!=3: continue # Tip 7
                        if nummer[gruen]!=nummer[weiss]-1: continue # Tip 4
                        if abs(nummer[marlboro]-nummer[katze])!=1: continue # Tip 10
                        if abs(nummer[dunhill]-nummer[pferd])!=1: continue # Tip 11
                        if abs(nummer[norweger]-nummer[blau])!=1: continue # Tip 13
                        if abs(nummer[marlboro]-nummer[wasser])!=1: continue # Tip 15

                    print
                    print "brit,schw,dae,nor,deu
",brite,schwede,daene,norweger,deutscher
                    print "hund,vogel,katze,pferd,fisch
",hund,vogel,katze,pferd,fisch
                    print "tee,kaffee,milch,bier,wasser
",tee,kaffee,milch,bier,wasser
                    print "pallmall,marlboro,dunhill,winfield,rothmans
",pallmall,marlboro,dunhill,winfield,rothmans
                    print "rot,gruen,weiss,gelb,blau
",rot,gruen,weiss,gelb,blau
                    print "nummern                                ",nummer[1:]
```

Das geht jetzt wie der Blitz!

Übung:

- Ergänze jedes der Programme um eine Zählvariable, wie oft der innerste Schleifenkörper abgearbeitet wird und zeige sie am Ende an. Erkläre das die Laufzeitunterschiede?
- Erfinde selbst derartige Rätsel! Ein Computerprogramm der obigen Art hilft dann bei der Kontrolle, dass es tatsächlich nur eine einzige (eindeutige) Lösung gibt!
- Eine Variation: erfinde mehrdeutige Rätsel, bei denen allerdings die Beantwortung der gestellten Frage eindeutig ist (Bsp: Ob der Brite im grünen oder weißen Haus wohnt, ist nicht zu entscheiden. Allerdings trinkt er mit Sicherheit Ginger Ale, und nach dem Getränk fragst Du dann auch.)