Lösungen der Aufträge

Lösung Auftrag 1. Finde meine Zahl (adversarial)

a) Spieler 2 kann nie die Anzahl Möglichkeiten genau halbieren, da er ungerade Zahlen hat: Im ersten Zug muss er eine Frage stellen, die mit ja 33 Zahlen entspricht und mit nein 32 Zahlen entspricht, oder umgekehrt. Wenn Spieler 1 adversarial spielt, bleiben nach dem ersten Zug 33 Möglichkeiten. Mit dem zweiten Zug teilt Spieler 2 die Zahlen 17 : 16 und Spieler 1 wählt den Teil mit 17 Zahlen. Nach Zug 3 werden die Zahlen 9 : 8 geteilt und der Teil mit 9 Zahlen wird gewählt. Nach Zug 4 bleiben 5 Zahlen übrig, 3 nach Zug 5, 2 nach Zug 6. 7 Züge sind nötig, um die Zahl zu finden.

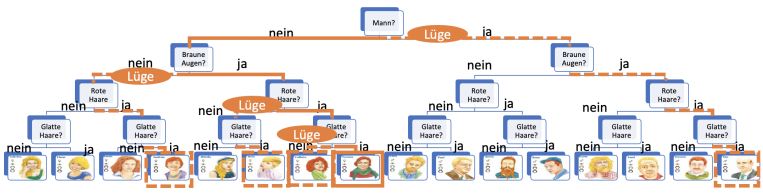
b) Wenn Spieler 1 sich einmal irrt und die kleinste Hälfte auswählt, braucht Spieler 2 nur noch 6 Züge, um die Zahl zu finden.

Lösung Auftrag 2. Einmal Lügen spielen

a) Um eindeutig die Antwort zu wissen, sollte eine Frage höchstens dreimal wiederholt werden: Die erste Wiederholung, um zu überprüfen, ob gelogen wurde. Die zweite Wiederholung ist nur nötig, wenn gelogen wurde, um zu überprüfen, welche Antwort richtig war.
Bemerkung: Wenn die zwei ersten Antworten nicht übereinstimmen, könnte die Lüge sowohl in der Antwort zur ersten Frage stattgefunden haben als auch in der Antwort zur wiederholten Frage.

b) Für jede der 4 Fragen muss man die Frage wiederholen, um die Antwort zu überprüfen. Wenn die Antworten nicht übereinstimmen, muss die Frage ein zweites Mal wiederholt werden. Im schlimmsten Fall braucht man insgesamt 9 Fragen.

Lösung Auftrag 3. Teilbaum mit einmal Lügen



Yvonne	0111
Callista	0110
Nele	0101
Gudrun	0011
Tim	1111

Lösung Auftrag 4. Selbstkorrigierende Kodierung mit Wiederholung der Eigenschaften

a) Binärzahl: 1111 (kein Fehler bei der Übermittlung)

b) Binärzahl: 1001 (Fehler in der zweiten Stelle, gemäss des Paritätsbit muss dieses Bit null sein)

c) Binärzahl: 1010 (Fehler im Paritätsbit)

d) Zwei Fehler in der Kodierung: Wir können die Zahl nicht entdecken.

Lösung Auftrag 5. Knobbelaufgabe: Entzifferung einer 9-Bits Kodierung

```
def parity(zahl):
    par = 0
    while zahl!=0:
        if zahl%2==1:
            par+=1
        zahl >>= 1
    return par % 2

# Diese Funktion funktioniert nur, wenn höchstens ein Fehler vorhanden ist
def decode(kodierung):
    zahl1 = kodierung >> 5
    zahl2 = (kodierung >> 1) & 0b1111
    if zahl1!=zahl2:
        if parity(zahl1)==kodierung%2:
            return zahl1
        else:
            return zahl2
    else:
        return zahl1

print(bin(decode(0b101110001)))
```

Lösung Auftrag 6. "Einmal Lügen": Die beste Strategie finden

a) Die Strategie funktioniert nicht, weil bei der Kontrollfrage eine Lüge auftreten kann. Deshalb kann das Gesicht, das den ersten 4 Antworten entspricht, nicht entfernt werden . Deshalb bleiben nach der ersten Kontrollfrage noch 5 Möglichkeiten übrig, und nicht 4. Infolgedessen bräuchte man noch 3 zusätzlichen Fragen und nicht 2, um mit 100%-iger Sicherheit die Lösung zu finden.

b) Die folgende Strategie findet die Lösung immer in 7 Zügen:
Du stellst erst die 4 Grundfragen f_1, f_2, f_3, f_4 zu den 4 Eigenschaften (zum Beispiel: f_1: *Bist du ein Mann?*, f_2: *Hast du braune Augen?*, f_3: *Hast du rotes Haar?*, f_4: *Hast du glattes Haar?*.
Du stellst dann 3 Kontrollfragen:

1. Die erste Kontrollfrage c_1 überprüft, ob alle Antworten zu den ersten 3 Grundfragen stimmten (zum Beispiel: *Stimmt es, dass du ein Mann mit braunen Augen und rotes Haar bist?*).
2. Die zweite Kontrollfrage c_2 überprüft, ob alle Antworten zu den Grundfragen f_1, f_2, f_4 stimmten.
3. Die dritte Kontrollfrage c_3 überprüft, ob alle Antworten zu den Grundfragen f_1, f_3 und f_4 stimmten.

Mit diesen Fragen kann man genau bestimmen, ob gelogen wurde und gegebenenfalls in welcher Frage gelogen wurde, auch wenn die Lüge in einer der Kontrollfragen stattgefunden hat.

c) Anstatt der oben erwähnten Kontrollfragen kann man eine Frage zur Parität der betroffenen Grundfragen stellen.

Lösung Auftrag 7. 7-Bits Selbstkorriegierende Kodierung

a) 1101 (keinen Fehler)

b) 1001 (Fehler im zweiten Bit)

c) 0011 (Fehler in der ersten Kontrollfrage)