

Aufgabe 4: Nandu

Team-ID: 01004

Team-Name: Was?

Bearbeiter dieser Aufgabe:

Mathieu de Borman

19. November 2024

Inhaltsverzeichnis

1 Lösungsidee	1
2 Umsetzung	1
3 Beispiele	2
4 Quellcode	5

1 Lösungsidee

Teilaufgabe 1

Es wird durch jede mögliche Kombination durchgegangen, indem man im Binärsystem runterzählt.

Teilaufgabe 2

Bei Jeder Kombination wird zuerst bei der ersten Zeile eingesetzt, ob sie an oder aus sind. Anschließend wird jede Zeile durchgegangen. Wenn die Stelle kein Block hat, dann ist sie aus. Wenn die Stelle Blau ist, dann wird einfach der gleiche Helligkeitswert wie der obendrüber eingesetzt. Wenn die Stelle Weiß ist, dann wird wenn die Beiden Stellen über dem Weißen an sind die beiden aus. Sonst sind sie an. Und bei Rot geht die Stelle an wenn der Wert über der Öffnung aus ist. In der Letzten Zeile wird immer der Wert obendrüber benutzt um den Output L zu haben

2 Umsetzung

Die Lösungsidee wird in JavaScript implementiert. Zum einen gibt es die Funktion austesten, die einmal die Kombination durchsimuliert. Und eine for Schleife die alle verschiedenen Kombinationen durchprobiert. Die Funktion hat einen Input. Der Input soll die zu simulierende Kombination sein. In der ersten for Schleife der Funktion wird dann bei jedem Input der Klötzchen Struktur eingetragen ob sie an oder aus ist. Das wird dann auch direkt in die Tabelle eingetragen. In den darauf folgenden beiden, ineinander geschachtelten for Schleifen, wird durch jede Zeile und dann durch jede Stelle gegangen. Wenn die Stelle ein X ist, dann wird die Stelle als aus gemerkt. Wenn die Stelle ein B ist, dann wird genau der wert wie der der Stelle obendrüber gemerkt. Wenn die Stelle ein W ist, dann wird überprüft, ob der Block obendrüber und der obendrüber, rechts daneben an sind. Wenn beide an sind, dann werden die beiden weißen Blöcke als aus eingetragen. Die nächste Stelle bekommt den

gleichen wert und wird deswegen übersprungen. Wenn es R ist, dann wird das Gegenteil des Lichtverhältnis des Feldes obendrüber bei sich und dem Feld neben dem R eingetragen das Feld rechts neben R wird übersprungen. Bei r wird das gleiche wie bei R gemacht mit dem Unterschied, dass nicht das Feld über der aktuellen Position überprüft wird sondern das Feld rechts neben dem Feld der aktuellen Position. Nachdem durch alle Zeilen gegangen wurde, trägt der code innerhalb der nächsten for Schleif das Ergebnis in die Tabelle ein. Vor der for Schleife, die alle fälle ausprobiert, wird jeder Input als eine Weitere eins in eine Binärzahl eingetragen. Das Ergebnis dessen sind dann die verschiedenen Fälle die alle durchprobiert werden. Die for Schleife Nimmt diese Zahl und macht sie mit jedem Schritt um eins kleiner. Bedeutet: solange es noch eine Kombination gibt, wird die for Schleife noch ein weiteres mal ausgeführt. In der for Schleife wird dann die Zahl wieder in binär umgewandelt. Diese wird dann wiederum geteilt und in eine Liste eingespeist. In der While Schleife wird dann, solange die Liste nicht alle fälle abdeckt am Anfang der Liste noch eine 0 angehängt. In der for Schleife nach der While Schleife, wird dann das 1 und 0 der Liste in true und false umgetauscht. Anschließend wird die Liste als Input für die Funktion gegeben.

3 Beispiele

Wir rufen nun das JavaScript-Programm mit den verschiedenen BWINF-Eingabedateien auf. Diese Dateien liegen im Selben Ordner wie die Programmdatei. Das Programm wird mit Hilfe des Browsers ausgeführt. In dem Dialogfeld kann man dann die Struktur eingeben.

Lösungen:

nandu1.txt

Q1 Q2 L1 L2

Aus Aus An An
 Aus An An An
 An Aus An An
 An An Aus Aus

nandu2.txt

Q1 Q2 L1 L2

Aus Aus Aus An
 Aus An Aus An
 An Aus Aus An
 An An An Aus

nandu3.txt

Q1 Q2 Q3 L1 L2 L3 L4

Aus Aus Aus An Aus Aus An
 Aus Aus An An Aus Aus Aus
 Aus An Aus An Aus An An
 Aus An An An Aus An Aus
 An Aus Aus Aus An Aus An
 An Aus An Aus An Aus Aus
 An An Aus Aus An An An

Q1 Q2 Q3 L1 L2 L3 L4

An An An Aus An An Aus

nandu4.txt

Q1 Q2 Q3 Q4 L1 L2

Aus Aus Aus Aus Aus Aus

Aus Aus Aus An Aus Aus

Aus Aus An Aus Aus An

Aus Aus An An Aus Aus

Aus An Aus Aus An Aus

Aus An Aus An An Aus

Aus An An Aus An An

Aus An An An An Aus

An Aus Aus Aus Aus Aus

An Aus Aus An Aus Aus

An Aus An Aus Aus An

An Aus An An Aus Aus

An An Aus Aus Aus Aus

An An Aus An Aus Aus

An An An Aus Aus An

An An An An Aus Aus

nandu5.txt

Q1 Q2 Q3 Q4 Q5 Q6 L1 L2 L3 L4 L5

Aus Aus Aus Aus Aus Aus Aus Aus Aus An Aus

Aus Aus Aus Aus Aus An Aus Aus Aus An Aus

Aus Aus Aus Aus An Aus Aus Aus Aus An An

Aus Aus Aus Aus An An Aus Aus Aus An An

Aus Aus Aus An Aus Aus Aus Aus An Aus Aus

Aus Aus Aus An Aus An Aus Aus An Aus Aus

Aus Aus Aus An An Aus Aus Aus Aus An An

Aus Aus Aus An An An Aus Aus Aus An An

Aus Aus An Aus Aus Aus Aus Aus Aus An Aus

Aus Aus An Aus Aus An Aus Aus Aus An Aus

Aus Aus An Aus An Aus Aus Aus Aus An An

Aus Aus An Aus An An Aus Aus Aus An An

Aus Aus An An Aus Aus Aus Aus An Aus Aus

Aus Aus An An Aus An Aus Aus An Aus Aus

Aus Aus An An An Aus Aus Aus Aus An An

Aus Aus An An An An Aus Aus Aus An An

Aus An Aus Aus Aus Aus Aus Aus Aus An Aus

Aus An Aus Aus Aus An Aus Aus Aus An Aus

Q1	Q2	Q3	Q4	Q5	Q6	L1	L2	L3	L4	L5
Aus	An	Aus	Aus	An	Aus	Aus	Aus	Aus	An	An
Aus	An	Aus	Aus	An	An	Aus	Aus	Aus	An	An
Aus	An	Aus	An	Aus	Aus	Aus	Aus	An	Aus	Aus
Aus	An	Aus	An	Aus	An	Aus	Aus	An	Aus	Aus
Aus	An	Aus	An	An	Aus	Aus	Aus	Aus	An	An
Aus	An	Aus	An	An	An	Aus	Aus	Aus	An	An
Aus	An	An	Aus	Aus	Aus	Aus	Aus	Aus	An	Aus
Aus	An	An	Aus	Aus	An	Aus	Aus	Aus	An	Aus
Aus	An	An	Aus	An	Aus	Aus	Aus	Aus	An	An
Aus	An	An	Aus	An	An	Aus	Aus	Aus	An	An
Aus	An	An	An	Aus	Aus	Aus	Aus	An	Aus	Aus
Aus	An	An	An	An	Aus	Aus	Aus	Aus	An	An
An	Aus	Aus	Aus	Aus	Aus	An	Aus	Aus	An	Aus
An	Aus	Aus	Aus	Aus	An	An	Aus	Aus	An	Aus
An	Aus	Aus	Aus	An	Aus	An	Aus	Aus	An	An
An	Aus	Aus	Aus	An	An	An	Aus	Aus	An	An
An	Aus	Aus	Aus	An	An	An	Aus	Aus	An	An
An	Aus	An	Aus	Aus	Aus	An	Aus	Aus	An	Aus
An	Aus	An	Aus	Aus	An	An	Aus	Aus	An	Aus
An	Aus	An	Aus	An	Aus	An	Aus	Aus	An	An
An	Aus	An	Aus	An	An	An	Aus	Aus	An	An
An	Aus	An	Aus	An	An	An	Aus	Aus	An	An
An	Aus	An	Aus	Aus	Aus	An	Aus	Aus	An	Aus
An	An	Aus	Aus	Aus	An	An	Aus	Aus	An	Aus
An	An	Aus	Aus	An	Aus	An	Aus	Aus	An	An
An	An	Aus	Aus	An	An	An	Aus	Aus	An	An
An	An	Aus	An	Aus	Aus	An	Aus	An	Aus	Aus
An	An	Aus	An	Aus	An	An	Aus	An	Aus	Aus
An	An	Aus	An	An	Aus	An	Aus	Aus	An	An
An	An	An	Aus	Aus	Aus	An	Aus	Aus	An	Aus
An	An	An	Aus	Aus	An	An	Aus	Aus	An	Aus
An	An	An	Aus	An	Aus	An	Aus	Aus	An	An

Q1	Q2	Q3	Q4	Q5	Q6	L1	L2	L3	L4	L5
An	An	An	Aus	An	An	An	Aus	Aus	An	An
An	An	An	An	Aus	Aus	An	Aus	An	Aus	Aus
An	An	An	An	Aus	An	An	Aus	An	Aus	Aus
An	An	An	An	An	Aus	An	Aus	Aus	An	An
An	An	An	An	An	An	An	Aus	Aus	An	An

4 Quellcode

```

let bild = document.getElementById("bild")

let context = bild.getContext("2d")

var table = document.createElement("table")

var thead = document.createElement("thead") //erstelle eine Tabelle

var headerRow = document.createElement("tr")

let eingabe = prompt("Nandu")//öffne das Dialogfeld

let array2 = []

let teil = []

array = eingabe.split(" ")//teile die Eingabe in mehrere Werte einer Liste

for (let x of array) { //Filter die Leerzeichen heraus
    teil = x.split(" ")

    for (let y of teil) {
        if (y == " " || y == "") {
        } else {
            array2.push(y)
        }
    }
}

let zwodliste = []

let breite = array2.shift()//speicher die breite

let höhe = array2.shift()//speicher die höhe

let teilx = []

for (let z = 0; z < höhe; z++) { //für jede Spalte
    teilx = []

    for (let i = 0; i < breite; i++) { //für jede Stelle
        teilx.push(array2[0])//verlängere die spalte
    }
}

```

```

        array2.shift()
    }
    zwodliste.push(teilx)//füge die Spalte der 2Dimensionalen Liste hinzu
}

let eins = 1
for (let ui of zwodliste[0]) { //für jede stelle in der ersten Zeile
    if (ui != "X") { //wenn das Feld nicht leer ist
        var headerCell1 = document.createElement("th")
        headerCell1.textContent = "Q" + eins
        //füge in der ersten Tabellenzeile eine Zelle hinzu, in der die Nummer des Inputs steht
        headerRow.appendChild(headerCell1)
        thead.appendChild(headerRow)
        table.appendChild(thead)
        eins++
    }
}

let zwei = 1
for (let c of zwodliste[höhe - 1]) { //für jede stelle in der letzten Zeile
    if (c != "X") { //wenn das Feld nicht leer ist
        var headerCell1 = document.createElement("th")
        headerCell1.textContent = "L" + zwei
        //füge in der ersten Tabellenzeile eine Zelle hinzu, in der die Nummer des Outputs steht
        headerRow.appendChild(headerCell1)
        thead.appendChild(headerRow)
        table.appendChild(thead)
        zwei++
    }
}

```

```

function austesten(qinputs) {
  //erstelle die funktion austesten mit dem Parameter welche Inputs an und welche aus sind
  var tbody = document.createElement("tbody")
  var row1 = document.createElement("tr")
  tbody.appendChild(row1)//erstelle eine neue Zeile
  let tf = []//erstelle eine Liste in der steht ob das licht aus oder an ist
  let teilvon = []//erstelle eine liste in der steht ob das licht aus oder an ist für die erste Zeile
  let hochzahlen = 0
  for (let ui of zwodliste[0]) { //für jede Stelle in der ersten Zeile
    if (ui == "X") { //Wenn nichts ist
      teilvon.push(false)//dann ist das licht aus
    } else { //sonst
      var cell1 = document.createElement("td")
      if (qinputs[hochzahlen]) { //wenn der aktuelle Input an ist
        cell1.textContent = "An" //füge der Zeile eine Stelle in der an steht hinzu
      } else {
        cell1.textContent = "Aus" //füge der Zeile eine Stelle in der aus steht hinzu
      }
      row1.appendChild(cell1)
      teilvon.push(qinputs[hochzahlen])
    }
    //die lampe an dieser Stelle, ist so wie es am eingang der funktion angegeben wird
    hochzahlen++
  }
}

tf.push(teilvon) //füge die erste Zeile dem gesamten hinzu
let benutzt = []

for (let c = 1; c < höhe - 1; c++) { //für jede Zeile
  let stellemerker = 0
  teilvon = []//leere die daten für diese Zeile
  benutzt = []

```

```

for (let ui = 0; ui < breite; ui++) { //für jede stelle in dieser Zeile
    if (zwodliste[c][ui] == "X") { //wenn diese stelle leer ist
        teilvon.push(false) //Dann gibt es hier auch kein ausgegebenes Licht
    } else if (zwodliste[c][ui] == "B") {
        //wenn der Klotz an dieser Stelle Blau ist
        teilvon.push(tf[c - 1][stellemerker])
        //dann gebe die Lichtverhältnisse von oben nach unten weiter
    } else if (zwodliste[c][ui] == "W") {
        //wenn der Klotz an dieser Stelle Weiß ist
        if (tf[c - 1][stellemerker] && tf[c - 1][stellemerker + 1]) {
            //Wenn die beiden Stellen über dem weißen Klotz an sind
            teilvon.push(false) //dann ist der Weiße Klotz aus
            teilvon.push(false)
        } else { //wenn nicht beide Stellen über dem Klotz an sind
            teilvon.push(true) //dann ist der Weiße Klotz an
            teilvon.push(true)
        }
        stellemerker++ //überspringe die nächste Stelle da diese Garantiert auch Weiß ist
        ui++
    } else {
        if (zwodliste[c][ui] == "R") { //Wenn die Stelle Rot ist und den Lichtsensor hat
            if (tf[c - 1][stellemerker]) { //Wenn das Licht über dem Sensor an ist
                teilvon.push(false) //dann ist der Klotz aus
                teilvon.push(false)
            } else { //Wenn das Licht über dem Sensor aus ist
                teilvon.push(true) //dann ist der Klotz an
                teilvon.push(true)
            }
        } else { //Wenn die Stelle der rote Klotz ohne Lichtsensor ist
            if (tf[c - 1][stellemerker + 1]) { //Wenn das Licht über dem Sensor an ist
                teilvon.push(false) //dann ist der Rote Klotz aus
            }
        }
    }
}

```



```

        teilvon.push(false)
    } else { //Wenn das Licht über dem Sensor aus ist
        teilvon.push(true) //dann ist der Rote Klotz an
        teilvon.push(true)
    }
}
ui++ //überspringe die nächste Stelle da ich weiß, das sie auch Rot sein wird
stellemerker++
}
stellemerker++
}
tf.push(teilvon)

//füge diese Zeile dem gesamtbild hinzu(Erinnerung: tf speichert an welcher stelle etwas an ist und
an welcher nicht)
}
teilvon = []
let stellemerker = 0

for (let c = 0; c < breite; c++) { //für die gesamte breite der eingabe
    if (zwoedliste[höhe - 1][c] == "X") { //wenn in der letzten zeile auf der Breite nichts ist
        teilvon.push(false) //dann wird an dieser Stelle kein Licht produziert
    } else {
        teilvon.push(tf[höhe - 2][c])
    }
    //wenn dort etwas ist dann übernehm einfach die gleichen Lichverhältnisse von oben
    var cell1 = document.createElement("td");
    if (tf[höhe - 2][c]) { //wenn die stelle also an ist
        cell1.textContent = "An" //dann füge das der Auszugebenen Tabelle hinzu
    } else { //wenn dem nicht so ist
        cell1.textContent = "Aus"
    }
    //dann gebe in der Tabelle an, dass das licht bei diesem Output aus ist
}
row1.appendChild(cell1);

```

```

    }
    stellemerker++
} table.appendChild(tbody);
tf.push(teilvon)
}

let fälle = []
eins = eins - 1
let alleverschiedenen = 1 / 9
for (let stelle = 0; stelle < eins; stelle++) { //für alle Inputs der Klötzchenstruktur
    alleverschiedenen *= 10 //Mache die Binärzahl um eine Stelle länger
}
let binärvariable = 0
let binäliste = []
alleverschiedenen = parseInt(Math.floor(alleverschiedenen), 2)
//wandle die Binärzahl in eine Dezimalzahl um
for (let fälle = alleverschiedenen; fälle >= 0; fälle--) { //Für jede mögliche kombination
    binärvariable = fälle.toString(2) //Wandle die dezimalzahl in eine Binärzahl um
    binäliste = binärvariable.split("") //jede Ziffer der Binärzahl ist nun ein Wert in der Liste
    while (binäliste.length < eins) {
        //Wenn die Liste zu kurz ist, weil die Dezimalzahl einen zu kleinen wert hatte
        binäliste.unshift(0) //Füge am Anfang der Liste eine 0 hinzu
    }
    for (let r = 0; r < binäliste.length; r++) { //für jede Stelle in der Binärliste
        if (binäliste[r] == 0) { //wenn der Wert null ist
            binäliste[r] = true //dann muss das Licht dort an sein
        } else { //wenn der wert 1 ist
            binäliste[r] = false //dann muss das Licht dort aus sein
        }
    }
}
austesten(binäliste) //diese kombination testen wir dann einfach mal

```

```
}
```

```
document.body.appendChild(table) //fügt tabelle in das Dokument hinzu
```