# Aufgabe 5: Widerstand

# 5.1 Lösungsidee

### Modellierung einer Schaltung

Nehmen wir für einen Augenblick an, wir hätten bereits ausgewählt, welche Widerstände wir nutzen wollen und wie wir diese anordnen. Wie können wir diese Schaltung nun in unserem Programm darstellen und den Widerstandswert berechnen?

Wie in der Aufgabenstellung beschrieben, kann jede Schaltung in mehreren Schritten aufgebaut werden, wobei in jedem Schritt zwei Teilschaltungen verbunden werden. Für diese Verbindung gibt es zwei Möglichkeiten: Man kann die Widerstände entweder parallel oder in Reihe schalten.

Diese Struktur kann gut durch einen binären Baum dargestellt werden (Abbildung 5.1). Jedes Blatt stellt einen Widerstand aus der Grabbelkiste und jeder innere Knoten eine Verbindung dar. Der Widerstandswert jeder Teilschaltung lässt sich leicht aus den Widerstandswerten ihrer Kinder berechnen.

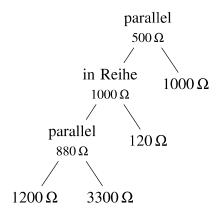


Abbildung 5.1: Darstellung des Beispiels aus der Aufgabenstellung als Baum.

### Struktur des Schaltungsbaums

Da die Schaltung aus maximal vier Widerständen bestehen soll, hat auch der Schaltungsbaum höchstens vier Blätter. Es gibt nur sehr wenige Möglichkeiten, einen Binärbaum mit bis zu vier Blättern zu konstruieren.

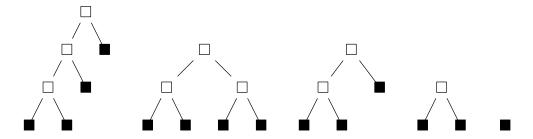


Abbildung 5.2: Alle möglichen Arten von Schaltungsbäumen.

Bäume, die durch Vertauschen der Kinder eines Knotens aus einem anderen Baum entstehen, brauchen wir nicht zu berücksichtigen, da es irrelevant ist, welcher von zwei Widerständen in einer Reihenschaltung *vorne* oder *hinten* bzw. in einer Parallelschaltung *oben* oder *unten* ist. Das ergibt sich daraus, dass die Formeln für die Berechnung des resultierenden Widerstands kommutativ sind, also a+b=b+a und  $\frac{1}{\frac{1}{a}+\frac{1}{b}}=\frac{1}{\frac{1}{b}+\frac{1}{a}}$  für zwei Widerstandswerte a und b.

Ein Sonderfall lässt sich nicht als Baum darstellen: die Schaltung, die nur aus einem Kabel ohne Widerstand besteht, mit einem Widerstandswert von  $0\,\Omega$ . Dieser lässt sich aber einfach getrennt behandeln.

Um alle möglichen Schaltungen auszuprobieren, müssen wir nur für die obigen *Baumarten* für jedes weiße Kästchen "parallel" und "in Reihe" und für jedes schwarze Kästchen alle möglichen Widerstände ausprobieren (Abbildung 5.2). Bei den Widerständen müssen wir darauf achten, dass sie nicht innerhalb einer Schaltung wiederverwendet werden können. In der Schaltung dürfen zum Beispiel nur dann zwei  $150\,\Omega$ -Widerstände vorkommen, wenn auch die Eingabe diesen zweimal enthält. Diejenigen Möglichkeiten, die duplizierte Widerstände enthalten, filtern wir also aus.

# 5.2 Optimierungen

#### Binäre Suche

Eine Möglichkeit zur Optimierung besteht in der Auswahl der Widerstände, die benutzt werden. Wir wählen erst mal nur drei statt vier Widerstände, sowie einen Platzhalter, und konstruieren daraus Bäume. Für jeden Baum können wir nun berechnen, was der optimale Widerstandswert für den Platzhalter wäre. Man sucht also ein *x*, sodass der Widerstand der Schaltung insgesamt *R* wird.

Im nächsten Schritt sucht man die beiden Widerstände, die am nächsten an x herankommen (ein Wert  $\geq x$  und einer  $\leq x$ ) und setzt diese an Stelle des Platzhalters ein. Eine dieser beiden Schaltungen wird das beste Ergebnis liefern, das mit den gewählten drei Widerständen und der Baumstruktur möglich ist. Mann muss also nur zwei Möglichkeiten für den vierten Widerstand ausprobieren. Dies funktioniert nur, da der Widerstandswert der Schaltung abhängig von x streng monoton steigend ist.

Wie berechnet man den Wert x? Am besten rekursiv: Sei v der Widerstand der Teilschaltung ohne Platzhalter. Wenn der aktuellen Baumknoten eine Reihenschaltung darstellt, muss die Teilschaltung mit Platzhalter den Widerstandswert R - v haben, um insgesamt R zu erreichen, bei einer Parallelschaltung stattdessen  $\frac{1}{\frac{1}{R} - \frac{1}{2}}$ .

### 5.3 Laufzeit

Sei *n* die Anzahl der Widerstände in der Grabbelkiste.

Der unoptimierte Algorithmus probiert für (fast) jede Auswahl von vier Widerständen aus dieser Kiste jeweils bis zu  $2^3$  Möglichkeiten (Zuweisungen von "parallel" und "in Reihe") für die fünf möglichen Baumstrukturen aus. Es gibt  $n^4$  Möglichkeiten, vier Widerstände aus der Kiste auszuwählen. Damit kommen wir auf bis zu  $2^3 \cdot 5 \cdot n^4 = 40n^4$  ausprobierte Bäume. Die Laufzeit des Algorithmus liegt also in  $\mathcal{O}(n^4)$ .

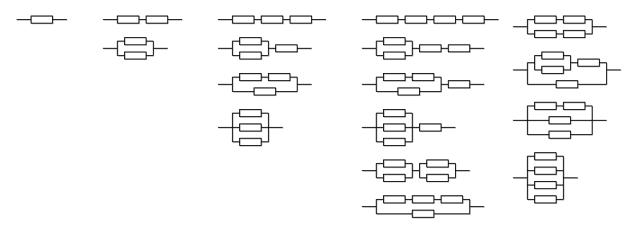
Der Algorithmus mit binärer Suche muss nur drei Widerstände aus der Kiste auswählen. Die binäre Suche zum Finden des vierten Widerstands läuft jeweils in  $\mathcal{O}(\log_2(n))$ . Dass wir am Anfang die Widerstände nach Wert sortieren müssen (typische Laufzeit in  $\mathcal{O}(n\log_2(n))$ ), fällt nicht weiter ins Gewicht. Insgesamt kommen wir auf eine Laufzeit von  $\mathcal{O}(n^3\log_2(n))$ .

Das ist bereits ein deutlicher Unterschied, wie man an den folgenden Laufzeitmessungen sieht:

Eingabe	n = 23	n = 50	n = 100
	Beispieleingabe	Zufallszahlen	Zufallszahlen
$\mathcal{O}(n^4)$ -Algorithmus	1.26s	27.24s	460.25s
$\mathcal{O}(n^3 \log_2(n))$ -Algorithmus	0.28s	2.69s	22.41s

# 5.4 Alternative Lösungsideen

Da die Anzahl der zu betrachtenden Schaltungen der bis zu k = 4 Widerstände noch überschaubar ist, können alternativ auch alle möglichen Schaltungen bis k = 4 betrachtet werden. Dabei kommen für k = 1, 2, 3, 4 die folgenden 1, 2, 4 bzw. 10 Schaltungen in Betracht:



Andererseits kann man auch vollkommen auf das Verwenden vorgegebener Schaltungsgraphen oder -bäume verzichten und die Schaltung komplett rekursiv durch die Parallel- und Reihenschaltung von vorhandenen und bereits zusammengebauten Widerständen aufbauen. Dabei müssen aber die Abbruchkriterien geeignet gewählt werden.

# 5.5 Beispiele

Für die vorgegebenen Beispiele berechnete das Programm die folgenden Ergebnisse:

### R = 500:

k = 1: (Widerstand 470  $\Omega$ )

k = 2: (Widerstand 500.380228  $\Omega$ )

k = 3: (Widerstand 500  $\Omega$ )

k = 4: (Widerstand 500  $\Omega$ )

### R = 140:

k = 1: (Widerstand 150  $\Omega$ )

k = 2: (Widerstand 140.425532  $\Omega$ )

k = 3: (Widerstand 139.949431  $\Omega$ )

k = 4: (Widerstand 140  $\Omega$ )

### R = 314:

k = 1: (Widerstand 330  $\Omega$ )

--| 330 |--

k = 2: (Widerstand 314.726508  $\Omega$ )

k = 3: (Widerstand 314.055637  $\Omega$ )

k = 4: (Widerstand 313.999343  $\Omega$ )

#### R = 315:

k = 1: (Widerstand 330  $\Omega$ )

--| 330 |--

k = 2: (Widerstand 314.726508  $\Omega$ )

k = 3: (Widerstand 315  $\Omega$ )

k = 4: (Widerstand 315  $\Omega$ )

#### R = 1620:

k = 1: (Widerstand 1500  $\Omega$ )

--| 1500 |--

k = 2: (Widerstand 1620  $\Omega$ )

--| 1500 |----| 120 |--

k = 3: (Widerstand 1620  $\Omega$ )

--| 1500 |----| 120 |--

k = 4: (Widerstand 1620  $\Omega$ )

--| 1500 |----| 120 |--

#### R = 2719:

k = 1: (Widerstand 2700  $\Omega$ )

--| 2700 |--

k = 2: (Widerstand 2700  $\Omega$ )

--| 2700 |--

k = 3: (Widerstand 2720  $\Omega$ )

--| 1800 |----| 100 |----| 820 |--

k = 4: (Widerstand 2719  $\Omega$ )

1. Runde

### R = 4242:

```
k = 1: (Widerstand 3900 Ω)

--| 3900 |--

k = 2: (Widerstand 4230 Ω)

--| 3900 |---| 330 |--

k = 3: (Widerstand 4240.7767 Ω)

|--| 390 |---|
-| | |---| 3900 |--
|--| 2700 |--|

k = 4: (Widerstand 4242 Ω)

|--| 120 |--|
-| | |---| 3900 |----| 270 |--
|--| 180 |--|
```

# 5.6 Bewertungskriterien

Die Bewertungskriterien (Fettdruck) vom Bewertungsbogen werden hier näher erläutert (Punktabzug in []).

- (1) [-1] Dokumentation sehr unverständlich bzw. unvollständig.
- (2) [-1] **Vorgegebenes Dateneingabeformat mangelhaft umgesetzt**:

  Die vorgegebenen Widerstände sollten aus der Datei korrekt eingelesen und geeignet abgespeichert werden. Es ist akzentabel wenn die Eingabe nur ganze Zahlen zulässt. Das

gespeichert werden. Es ist akzeptabel, wenn die Eingabe nur ganze Zahlen zulässt. Das Programm muss intern aber mit Gleitkommazahlen oder Brüchen rechnen, um für alle Schaltungen den korrekten Widerstandswert ermitteln zu können.

• (3) [-1] Lösungsverfahren fehlerhaft:

Das Lösungsverfahren sollte korrekt sein. Die einzelnen Widerstände sollten zwar nicht öfter als in der Grabbelkiste vorhanden in einer Schaltung verwendet werden, andernfalls wird aber auch kein Punkt abgezogen. Die Minimierung der Widerstandsdifferenz genügt, andere physikalisch sinnvolle Metriken werden nicht gefordert. Eine Berücksichtigung des Sonderfalls von 0 Widerständen bzw.  $0\,\Omega$  bleibt unbewertet, auch wenn er dokumentiert wurde.

• (4) [-1] Strategie für Schaltungsentwurf mangelhaft / fehlt:

Das Verfahren sollte immer das optimale Ergebnis zu berechnen versuchen und keinen heuristischen Ansatz folgen. Aufgrund der geringen Anzahl von Schaltungen ist eine Brute-Force-Strategie erlaubt. Eine naive Festverdrahtung aller möglichen Schaltungen durch Programmcode (statt der Verwendung einer Datenstruktur) führt zu Punktabzug, wenn sie mit einem erheblichen Anwachsen des Programmcodes einhergeht (d.h. nicht

nur eine deklarative Codezeile für jede mögliche Schaltung). Alle zehn möglichen Schaltungen, d. h. alle möglichen Reihen- und Parallelschaltungen der bis zu vier Widerstände sollten von der Strategie berücksichtigt werden.

### • (5) [-1] Verfahren bzw. Implementierung unnötig aufwendig:

Nur besonders umständliche Verfahren bzw. Implementierungen führen zu Punktabzug. Überlegungen zur Laufzeit, z. B in Abhängigkeit von der Anzahl verfügbarer Widerstände, werden nicht bewertet.

### • (6) [-1] Programmausgabe ungeeignet:

Die Programmausgabe sollte nachvollziehbar sein. Grafische Ausgaben sind schön, aber eine textuelle Ausgabe, welche die Struktur des Bauplans nachvollziehbar angibt, genügt. Sowohl der Bauplan der jeweils berechneten Schaltung mit ihren einzelnen Widerstandswerten als auch ihr Gesamtwiderstandswert R sollten ausgegeben werden. Sollte der geforderte Gesamtwiderstand R bereits mit weniger als 4 Widerständen in einer Schaltung möglich sein, so genügt ein Hinweis darauf. Die alleinige Angabe von Berechnungsformeln für den Gesamtwiderstand R ohne einen konkreten Bauplan führt zu Punktabzug.

### • (7) [-1] Beispiele fehlerhaft bzw. zu wenige:

Es wird die Dokumentation der Ergebnisse von mind. 2 aussagekräftigen Beispielen mit korrekten Ergebnisse erwartet, die die Güte des Programms sinnvoll demonstrieren. Wurden für k von 1 bis 4 Schaltungen berechnet, aber nur die beste Schaltung, d.h. die mit der geringster Widerstandsdifferenz zum gewünschten Gesamtwiderstand R, ausgegeben, wird kein Punkt abgezogen. Eigentlich war für jedes k von 1 bis 4 (bzw. bis der Wert R erreicht ist) die Ausgabe einer geeigneten Schaltung als Ergebnis erwartet worden, aber einige fanden die Aufgabenstellung in dieser Hinsicht nicht klar genug.