

# Grundlagen der Medieninformatik I

T12 - 26.11.2020

Kodierung



T12 - 26.11.2020

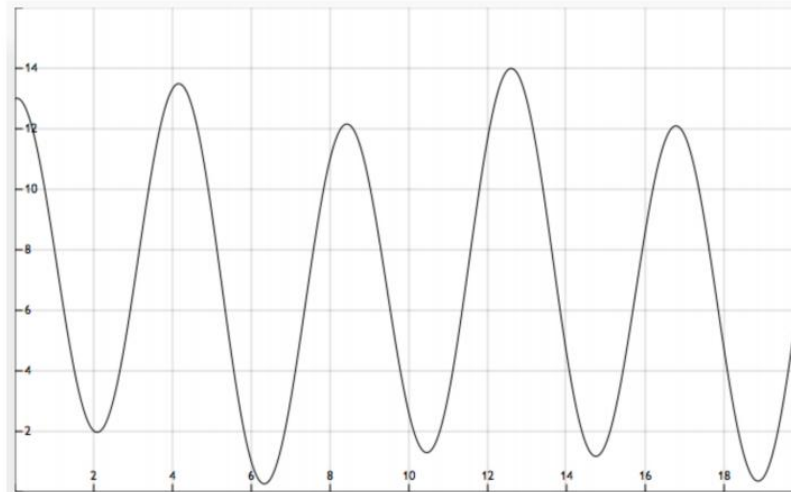
The image features the Kahoot! logo in a bold, white, sans-serif font. The text is centered horizontally and positioned in the middle of the frame. The background is a solid dark purple, with a lighter purple diagonal stripe running from the top-left towards the bottom-right, passing behind the text.

**Kahoot!**

# Übungsblatt 2 - Abgabe bis 29.11 23:59 GMT+1

## Übung 2: Digitalisierung

Einzelaufgabe, 10 Punkte, Abgabe 29.11.20, 23:59 Uhr in Stud.IP



**1. Signal digitalisieren:** Die Aufgabe ist es, obiges Signal angemessen zu digitalisieren:

- » Betrachte das Signal und wähle eine sinnvolle Samplingrate (mit Begründung). Die X-Achse ist Zeit in Sekunden, die Y-Achse hat willkürliche Einheiten. 1 P
- » Markiere die gesampleten Werte in der Grafik 1 P
- » Wähle eine sinnvolle Quantisierung und einen sinnvollen Wertebereich (mit Begründung) 1 P
- » Quantisiere die gesampleten Werte und stelle das Ergebnis als Folge von Dezimalzahlen dar 1 P

# Kodierung

- Binärdarstellung von Symbolen
- Ein digitales Signal wird durch Symbole dargestellt
- Jede Symbolmenge lässt sich binär kodieren
- $n$  Bits haben  $2^n$  Kombinationen (Hä? - Ja erklären wir Gleich)

# Kodierungsarten

- Welche Arten von Kodierung habt ihr gelernt?
  - Standardkodierung
  - Lauflängkodierung
  - Huffman Kodierung
  - Wörterbuchkodierung - Später in der Vorlesung



# Standartkodierung

- Kodierung mit fester Anzahl an Bits
- Verwende für alle Zeichen gleiche Anzahl von Bits
- Suche kleinste Zahl  $n$ , so dass  $2^n \geq |X|$
- $X$  - die Anzahl der zu kodierenden Zeichen
- z.B.  $X = \{A, B, C, D, E, F, G\} \rightarrow |X| = 7$ , suche  $2^n$  so dass  $2^n \geq 7$   
 $\rightarrow$  Kodiere daher mit 3 Bits, da  $2^3 = 8 > 7$

Kodiere z.B.

A  $\rightarrow$  000, B  $\rightarrow$  001 C  $\rightarrow$  010, D  $\rightarrow$  011, E  $\rightarrow$  100, F  $\rightarrow$  101 G  $\rightarrow$  110

# Standartkodierung

- Beispiel: Standartkodierung von "HALLO WELT" (Ohne Leerzeichen)
- $X = \{H, A, L, O, W, E, T\}$
- $|X| = 7 \rightarrow 2^3 = 8 > 7$
- Kodierung z.B.
- $H \rightarrow 000, A \rightarrow 001, L \rightarrow 010, O \rightarrow 011, W \rightarrow 100, E \rightarrow 101, T \rightarrow 110$
- Bekomme also (ohne Leerzeichen):

H	A	L	L	O	W	E	L	T
000	001	010	010	011	100	101	010	110

- "Hallo Welt" kodiert ist also: 000001010010011100101010110

# Standartkodierung

- Nachteil:
- Rohe, unkomprimierter Datenstrom
- → Mehr Speicherverbrauch
- Lösung:
- Komprimierung





# Lauf­läng­ko­die­rung

- Idee: Ersetzen einer Folge gleicher Zeichen durch 1 Zeichen + Zähler
- Nutze hierfür Steuerbits, z.B.  $\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow\rightarrow$
- z.B. kodierung von "HALLO WELT"
- Gleiches Verfahren wie vorher,  $|X| = 7 \rightarrow 2^3 = 8 > 7$
- $H \rightarrow 000, A \rightarrow 001, L \rightarrow 010, O \rightarrow 011, W \rightarrow 100, E \rightarrow 101, T \rightarrow 110$
- Bekomme somit: (SB - Steuer Bit)

Steuer-bits	Bedeutung
000	1 Zeichen direkt
001	2 Zeichen direkt
010	3 Zeichen direkt
011	4 Zeichen direkt
100	Zeichen 2× wdh.
101	Zeichen 3× wdh.
110	Zeichen 4× wdh.
111	Zeichen 5× wdh.

SB	H	A	SB	LL	SB	O	W	E	L	SB	T
001	000	001	100	010	011	011	100	101	010	000	110

- "Hallo Welt" ist damit: 001000001100010011011100101010000110

# Lauf­läng­ko­die­rung

- **Vorteil:**

- Weniger Bits bei Eingabe mit vielen Wiederholungen des gleichen Zeichens
  - Bei ABBBBEFAFFFFFFNFNNNNNN z.B. 51 Bits statt 60 Bits
- Datenkomprimierung

- **Nachteil:**

- Bei normaler Eingabe → Mehr Bits als mit Standardkodierung
- Bei “HALLO WELT” z.B. 36 Bits statt 27 Bits



# Huffman Kodierung

- Idee: Kodiere Zeichen einzeln, aber mit unterschiedlicher Anzahl an Bits, so dass häufige Zeichen kurze Bitfolgen haben
- Vorgehen:
  - Sortiere Zeichen nach Häufigkeit von klein zu groß
  - Verbinde jede 2 kleinst vorkommenden Zeichenknoten
  - Bis alle Knoten als Baum verbunden
- Setze dann noch 1en und 0en und bekomme kodierte Wort



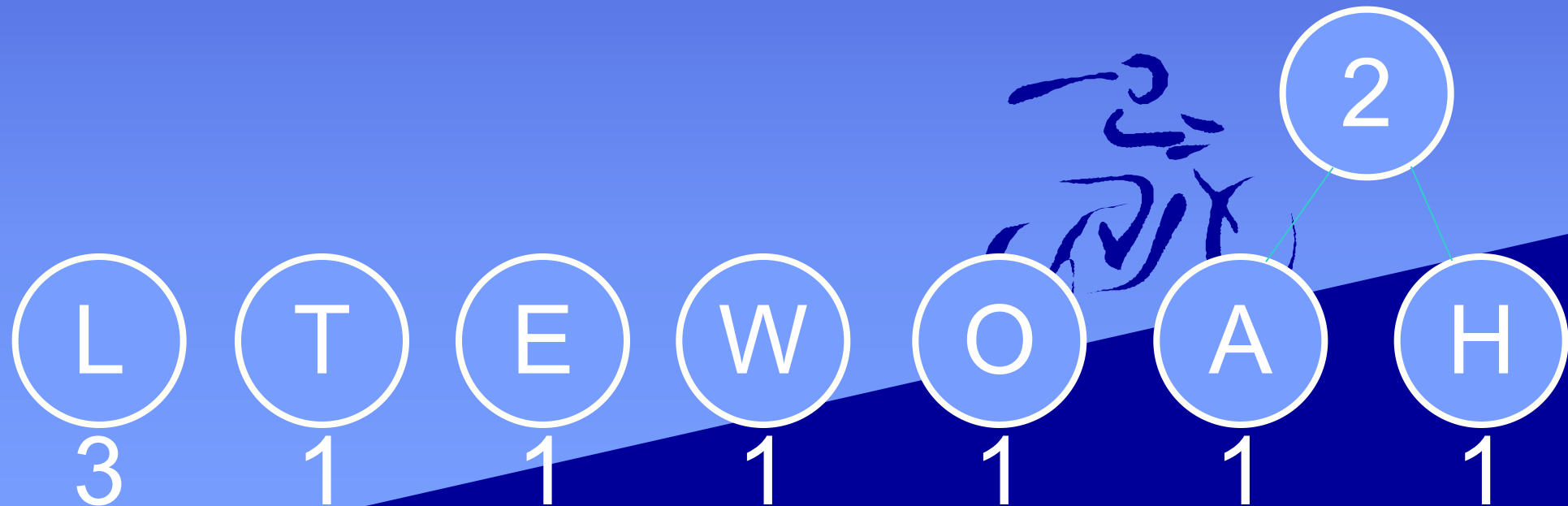
# Huffman Kodierung - Beispiel "HALLO WELT"

- \*\* Huffman Bäume können nach unterschiedlichen Verfahren erstellt werden, hier nutzen wir das folgende Verfahren: \*\*
- Schreibe Buchstaben auf nach Häufigkeit von Rechts nach Links (Groß- / Kleinschreibung hier ignoriert!)
- Verbinde kleinste Häufigkeiten zu gemeinsamem Knoten, bis alle Knoten als Baum verbunden sind



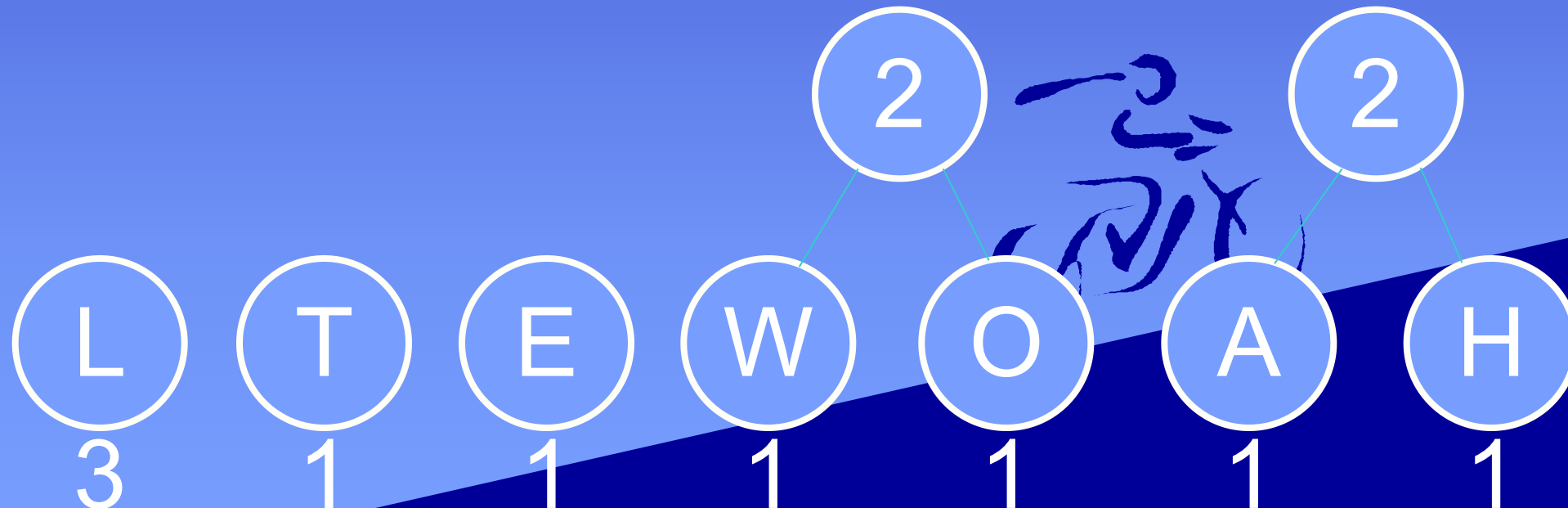
Häufigkeit:

# Huffman Kodierung - Beispiel "HALLO WELT"



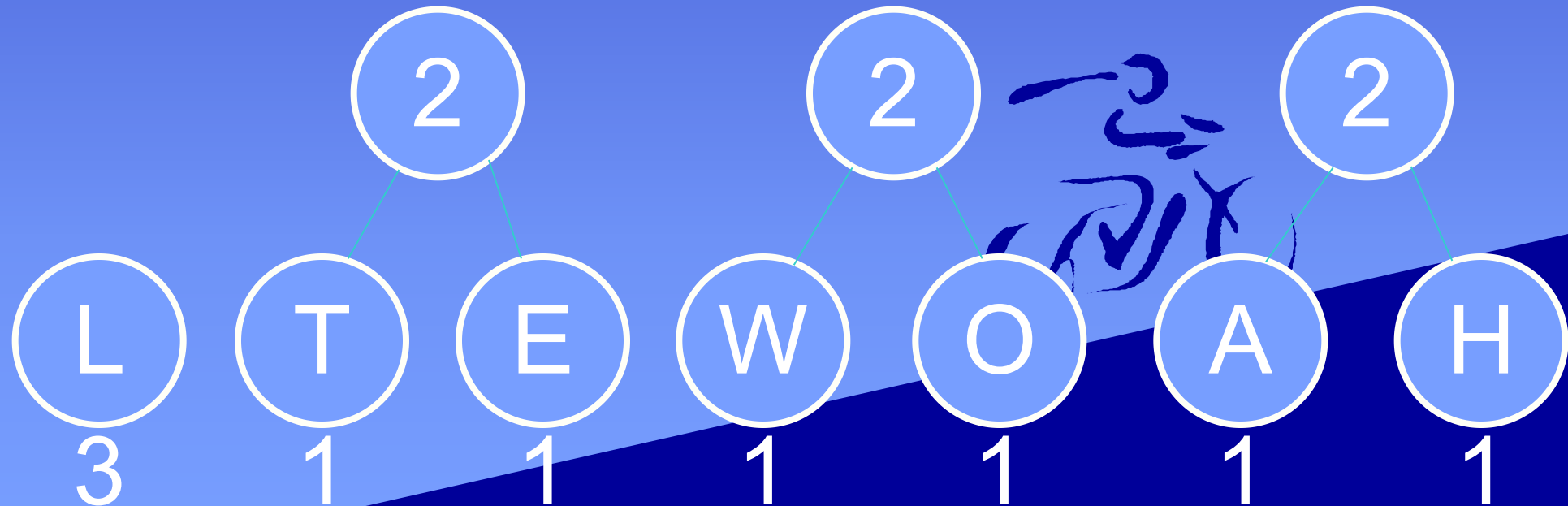
Häufigkeit:

# Huffman Kodierung - Beispiel "HALLO WELT"



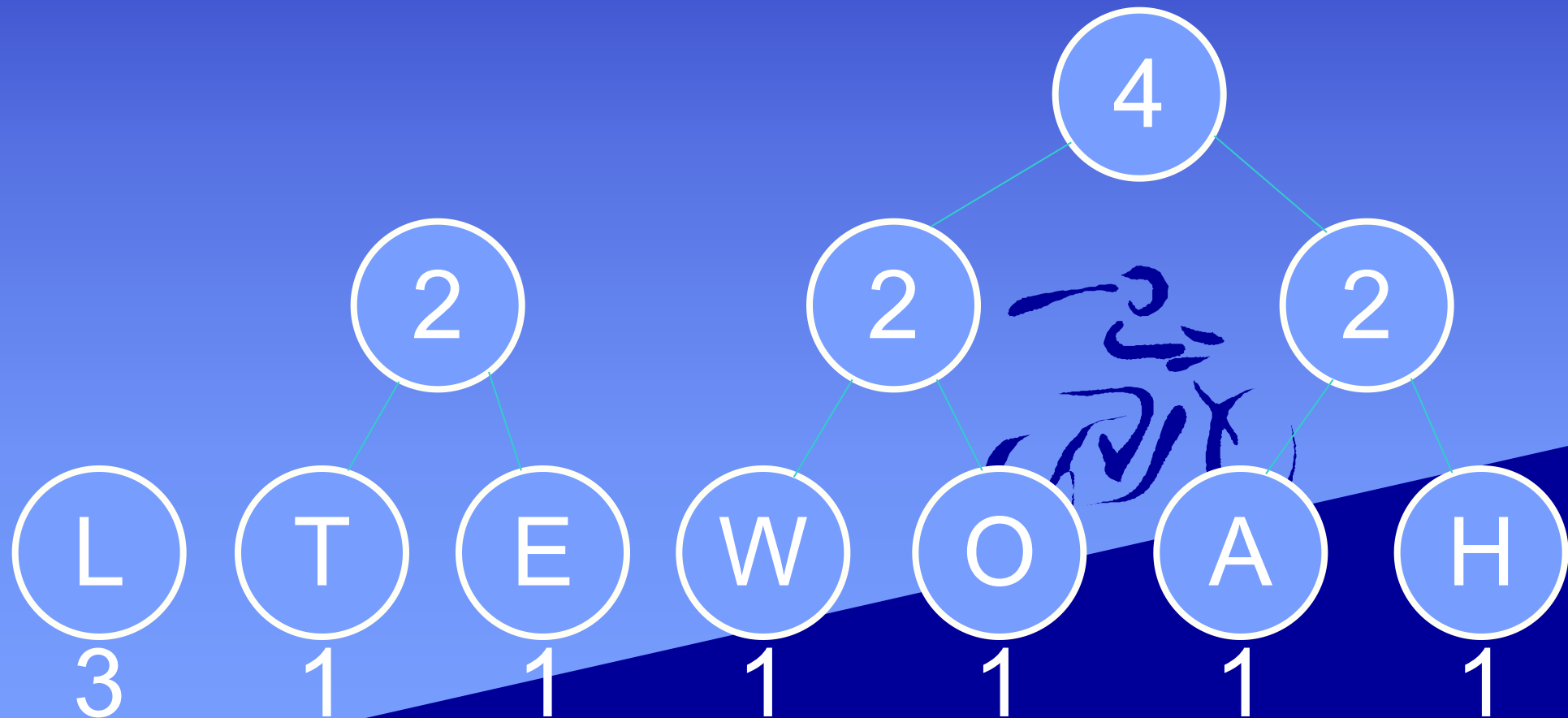
Häufigkeit:

# Huffman Kodierung - Beispiel "HALLO WELT"



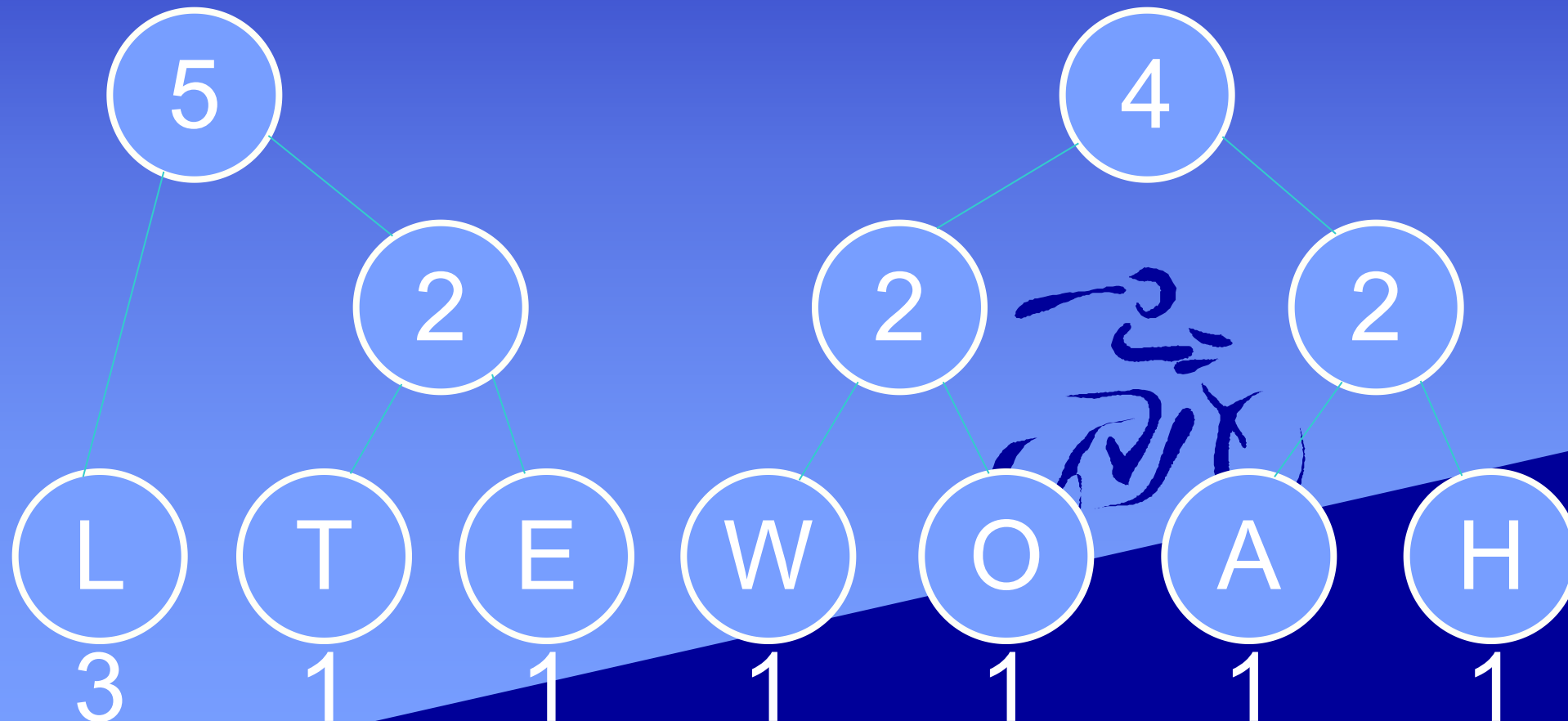
Häufigkeit:

# Huffman Kodierung - Beispiel "HALLO WELT"

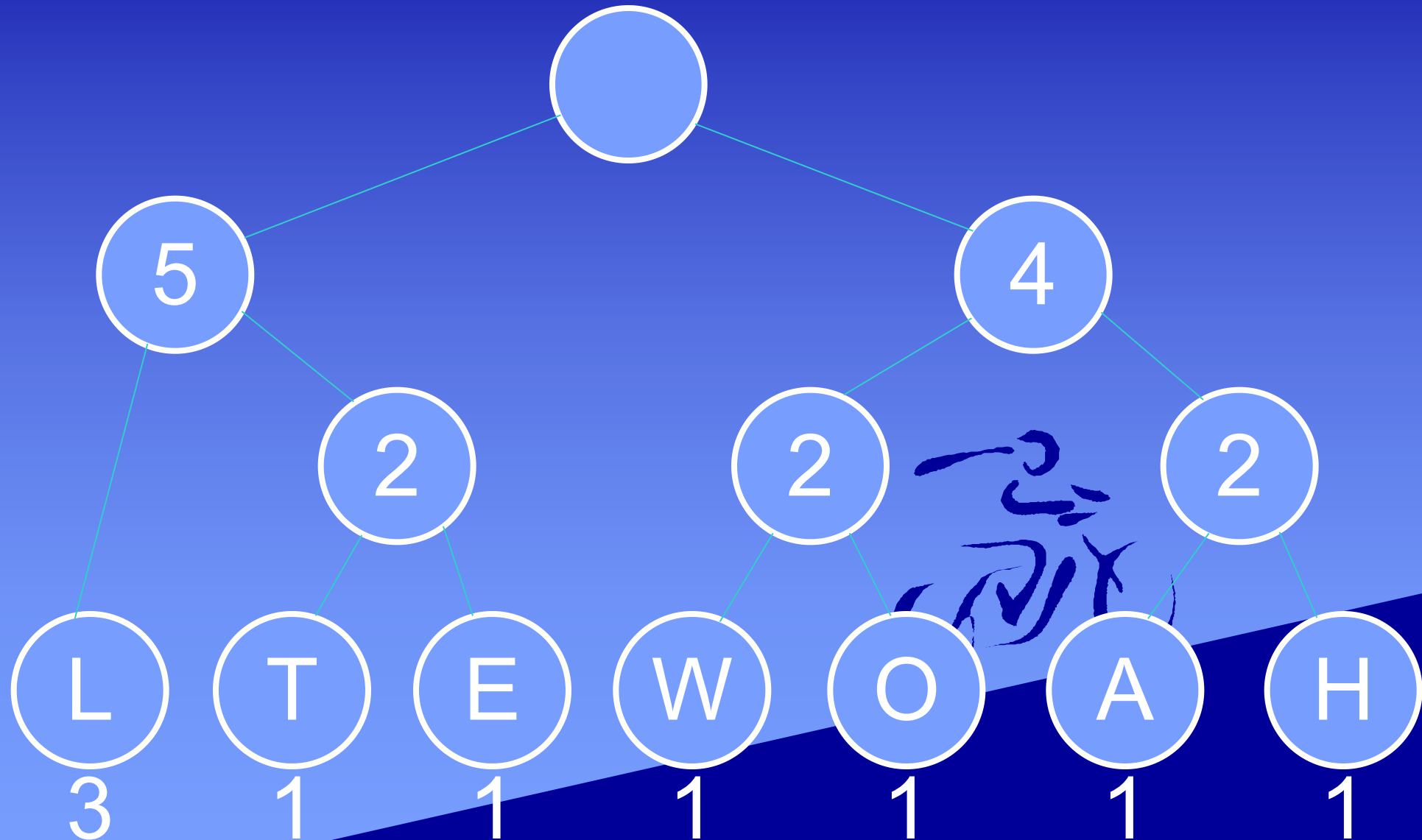




# Huffman Kodierung - Beispiel "HALLO WELT"

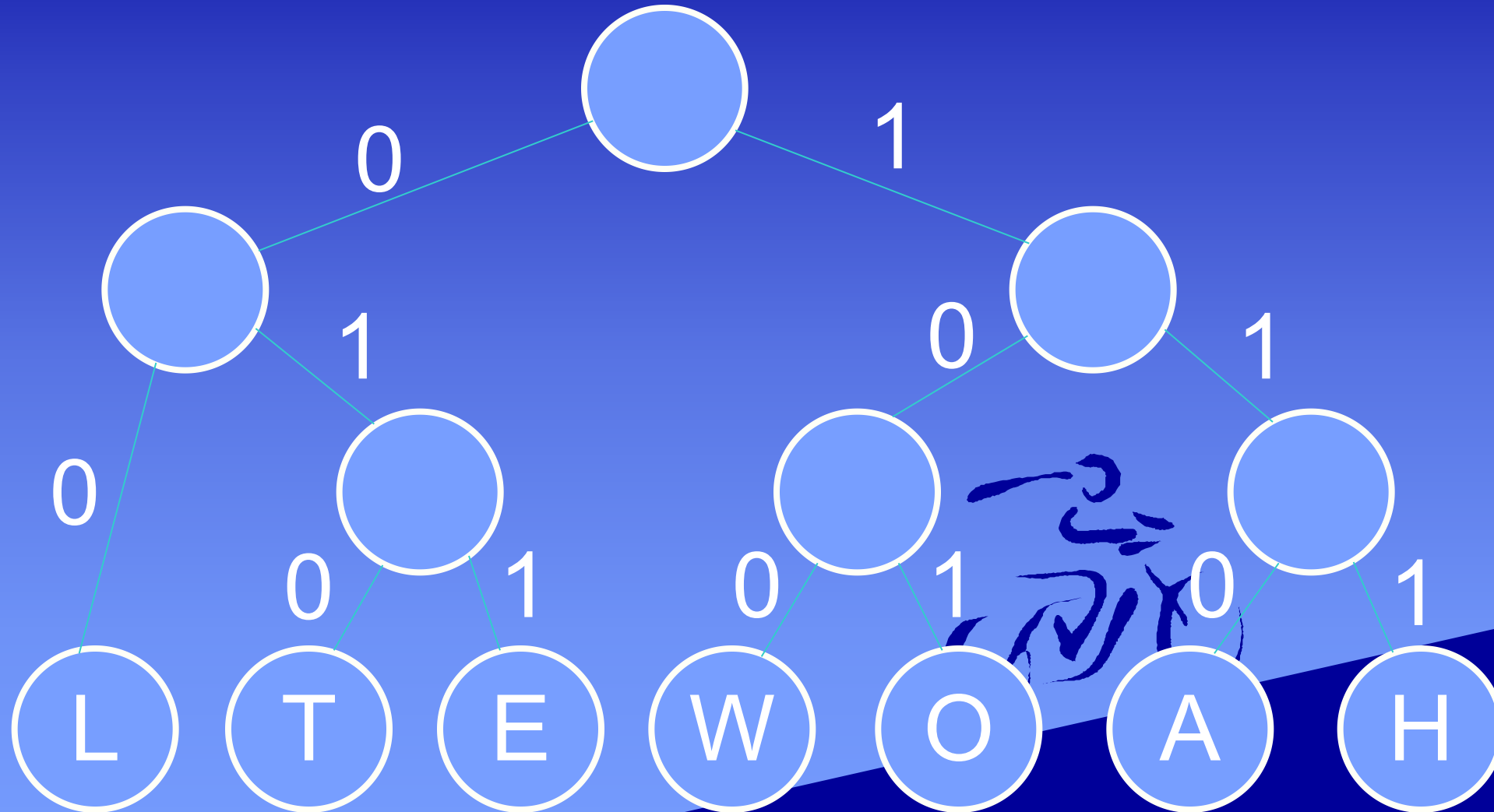


# Huffman Kodierung - Beispiel "HALLO WELT"



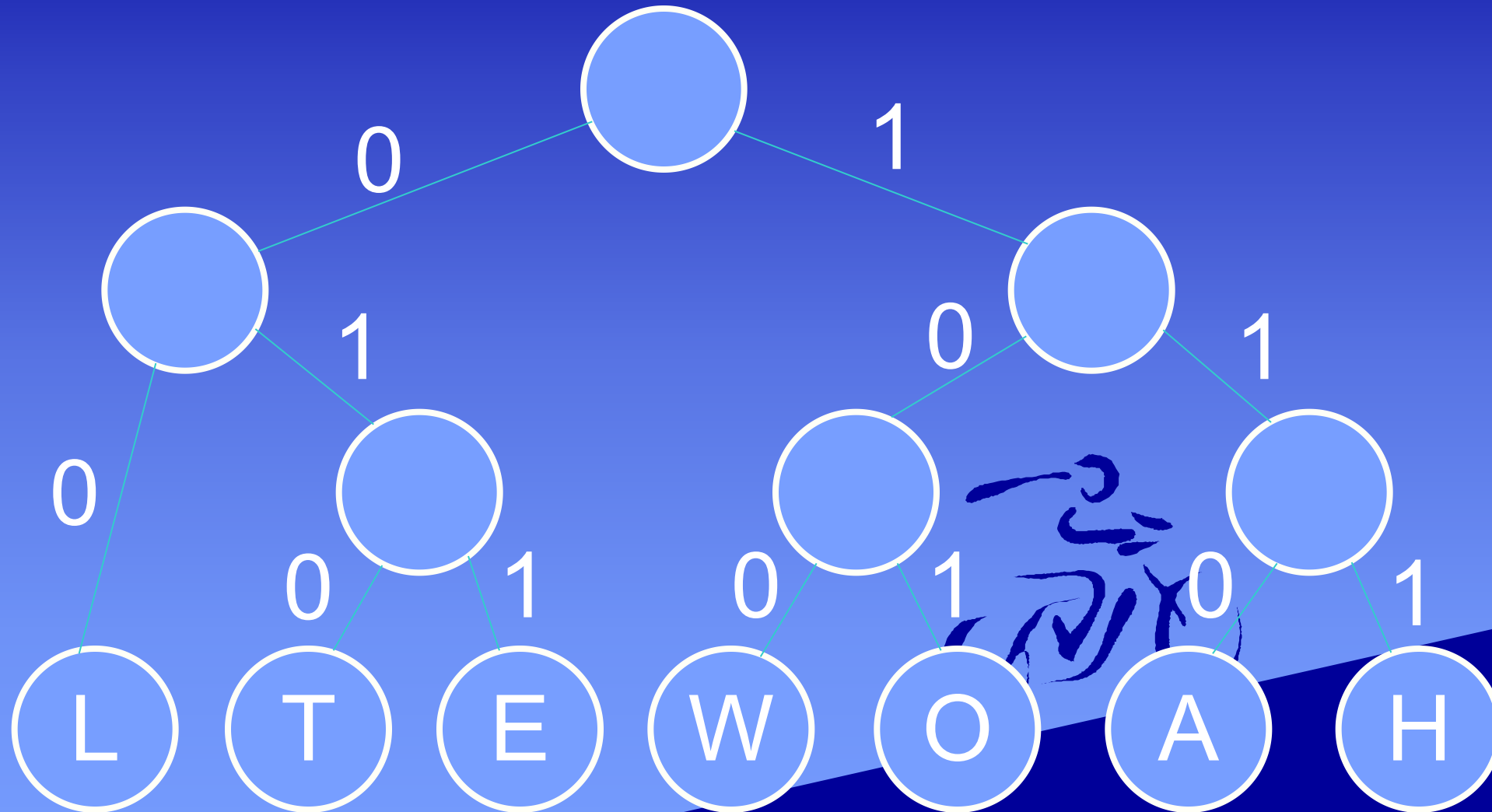
# Huffman Kodierung - Beispiel "HALLO WELT"

- Schreibe 1en Rechts, 0en Links



# Huffman Kodierung - Beispiel "HALLO WELT"

- Dekodiere...



# Huffman Kodierung - Beispiel "HALLO WELT"

- Lese Symbole aus dem Baum und bekomme:

H	A	L	L	O	W	E	L	T
111	110	00	00	101	100	011	00	010

- "Hallo Welt" = 11110000010110001100010
- Zur Überprüfung eurer Lösung kann ein Huffman Baum Generator verwendet werden. *Bei korrekter Lösung gleiche Bitanzahl (Baum kann vertauschte Knoten haben)*
- Bei einfachem Einsetzen in einen Generator und Copy Paste wird eure Abgabe mit 0 Punkten bewertet!

# Arbeitsblatt

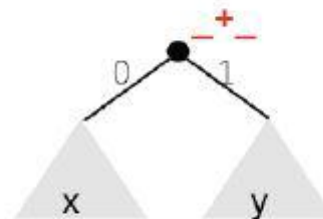
Kodiere die Eingabe

DABADABADU BALU

(ohne Leerzeichen) mit Hilfe der Huffman-Kodierung. Um wie viel reduziert sich die Bitzahl gegenüber einer Standard-Kodierung?

## Algorithmus zur Huffman – Kodierung

- Zähle in  $n(x)$  wie oft das Zeichen  $x$  in der Eingabe vorkommt
- Für jedes vorkommende Zeichen  $x$ 
  - füge einen Teilbaum mit einem Knoten hinzu, der nur das Zeichen  $x$  enthält und dessen Häufigkeit angibt
- Solange mehr als ein Teilbaum übrig ist
  - suche zwei Teilbäume  $x, y$  mit den zwei geringsten Häufigkeiten (ggf. wählen)
  - füge die Teilbäume  $x$  und  $y$  zusammen zu einem neuen Teilbaum, der
    - eine Wurzel hat,
    - $x$  im 0-Zweig der Wurzel hat,
    - $y$  im 1-Zweig der Wurzel hat und
    - als Häufigkeit die Summe der Häufigkeiten von  $x$  und  $y$  hat.
- Gehe Zeichen für Zeichen durch die Eingabe:

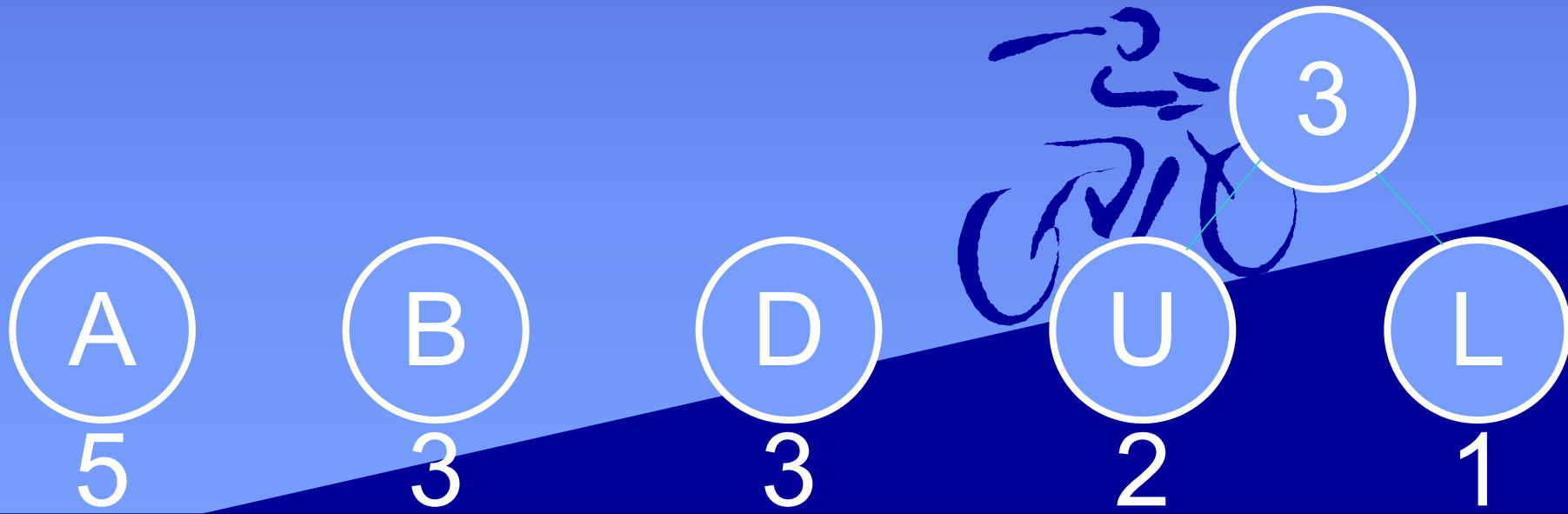


# Lösung

- Schreibe Zeichen nach Häufigkeit auf von Rechts nach Links
- Verbinde niedrigste Häufigkeiten zu Knoten...
- ...Bis alle Knoten als Baum verbunden sind

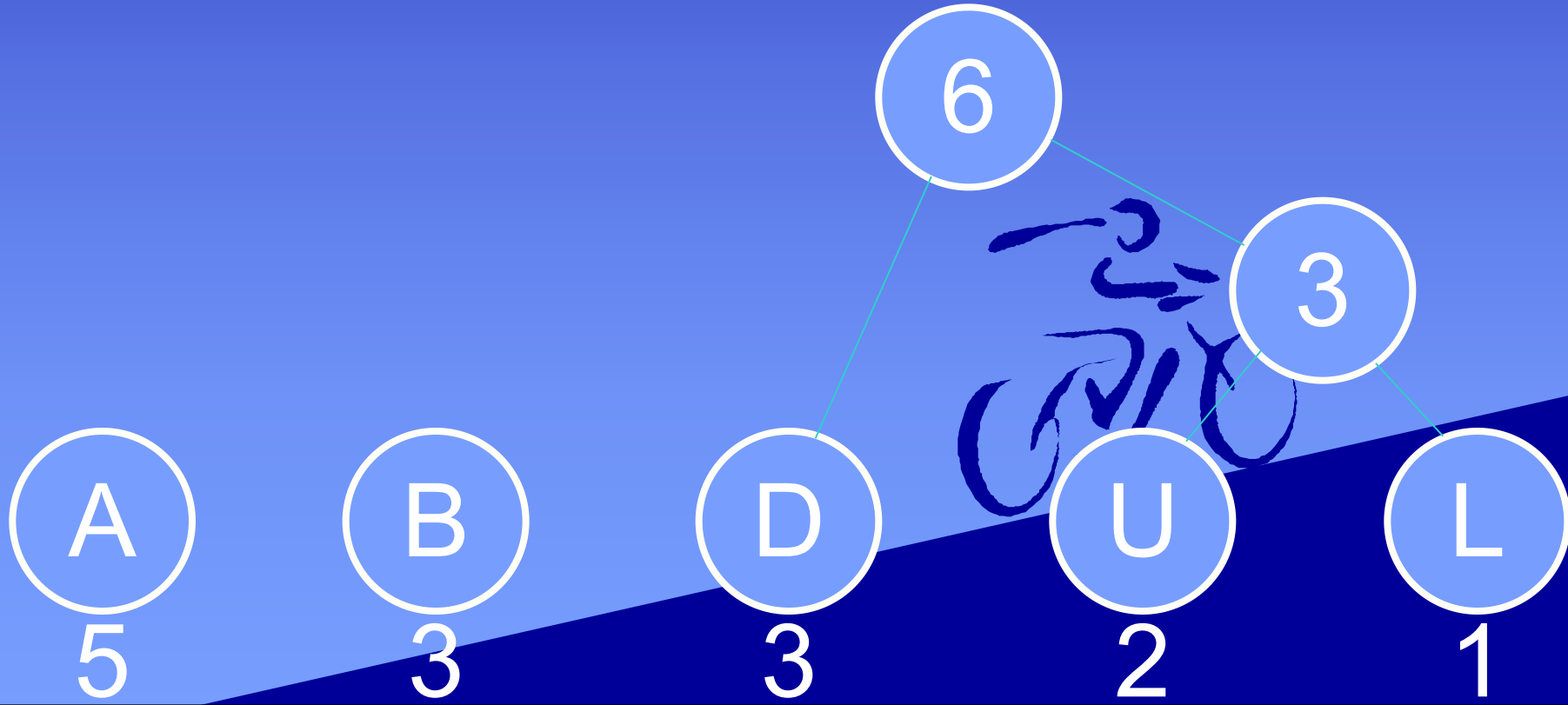


# Lösung

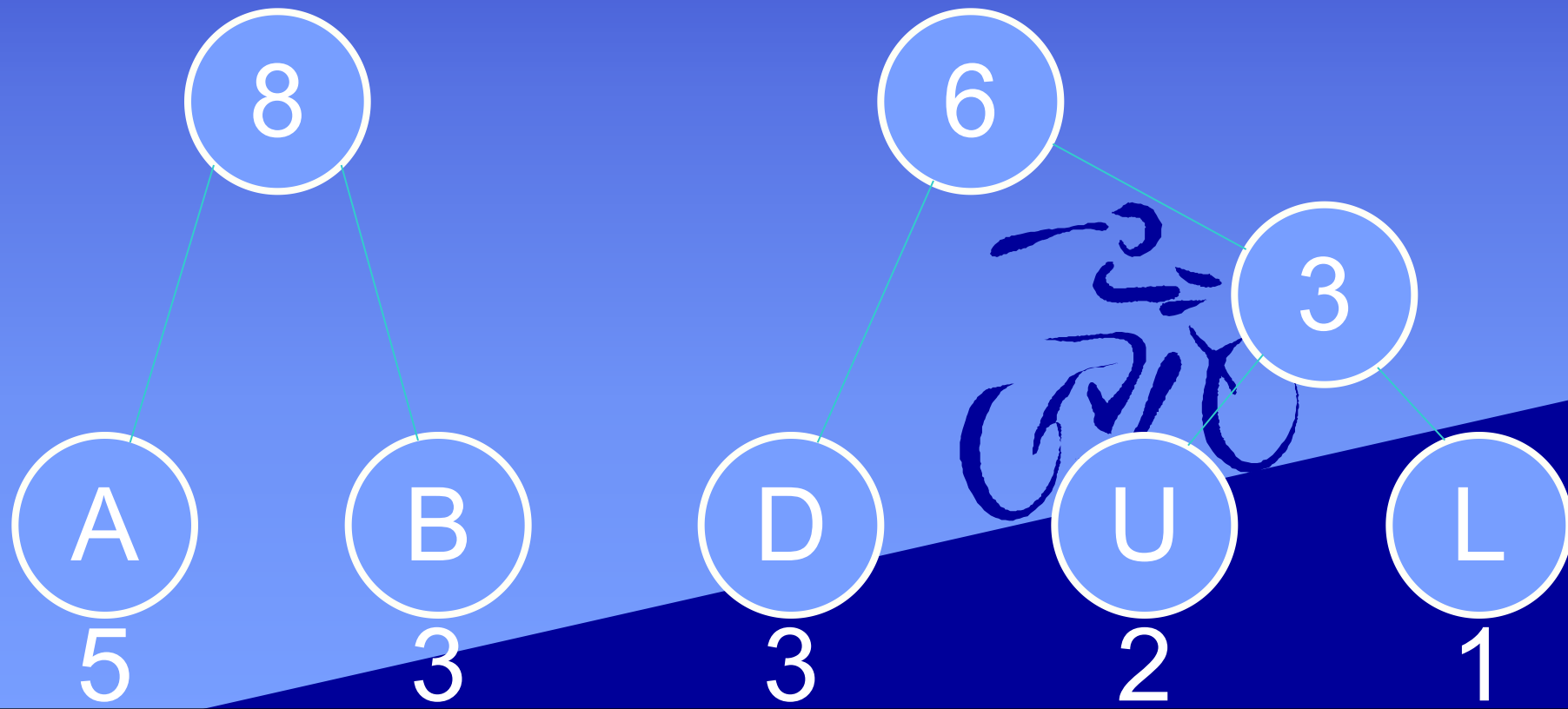




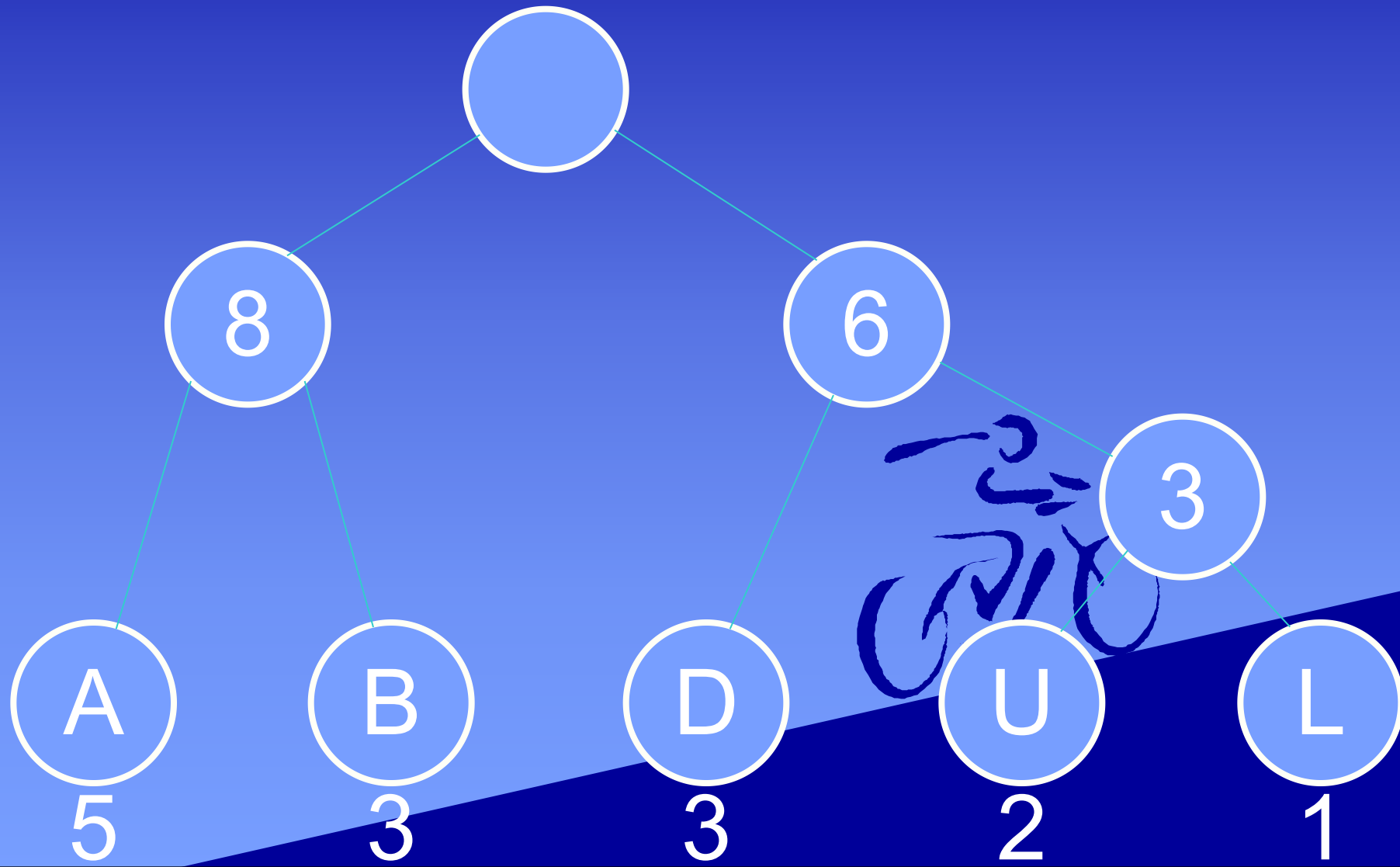
# Lösung



# Lösung

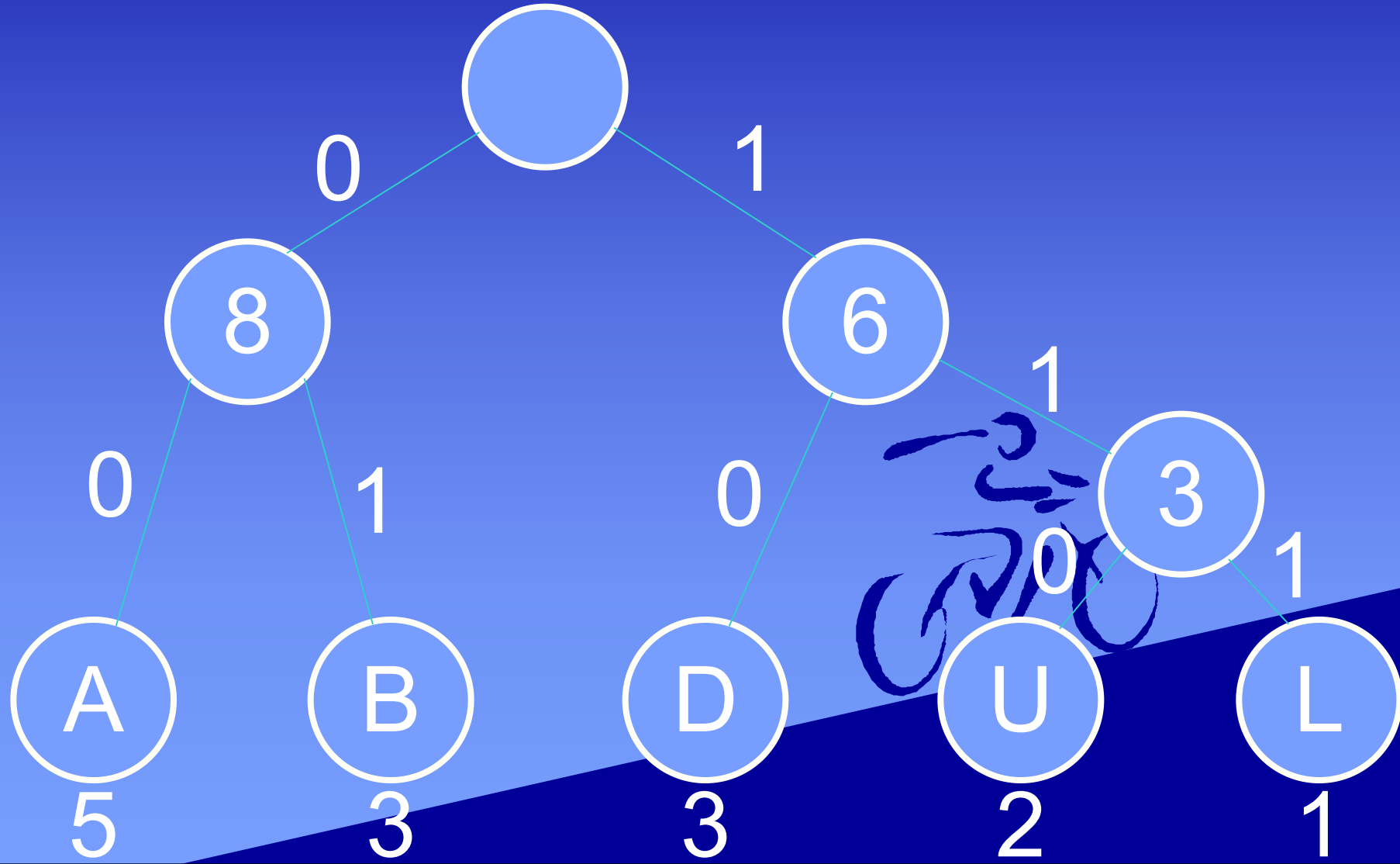


# Lösung



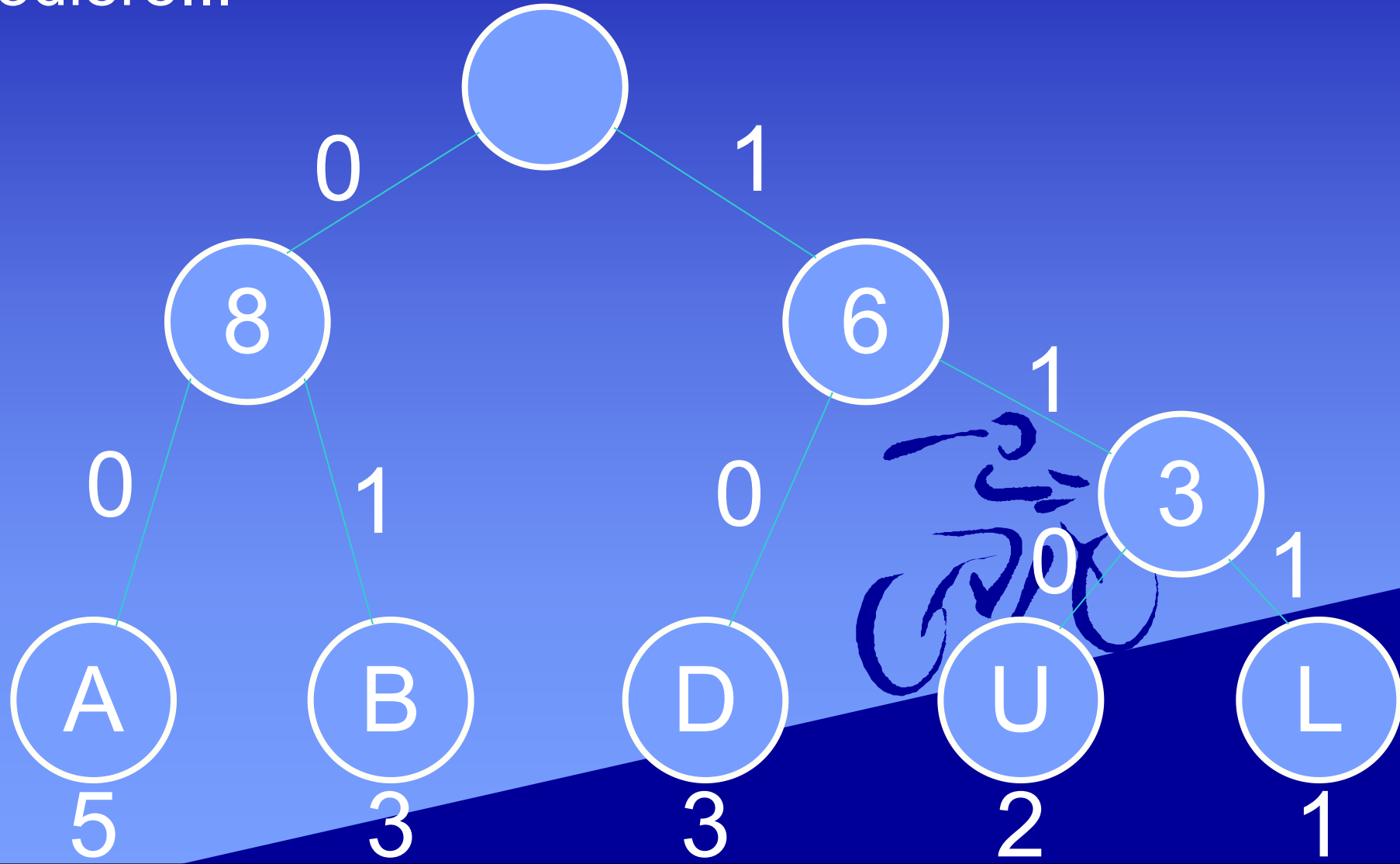
# Lösung

- 1en Rechts, 0en Links...



# Lösung

- Dekodiere...



# Lösung

- Lese die Symbole aus dem Baum und bekomme:

D	A	B	A	D	A	B	A	D	U	B	A	L	U
10	00	01	00	10	00	01	00	10	110	01	00	111	110

- Und somit ist DABADABADU BALU kodiert:
- = 1000010010000100101100100111110



# Lösung

- Reduzierung:
- Mit Standardkodierung hätte jedes Zeichen 3 Bit ( $2^3 > 5$ )
- Das Wort ist 14 Zeichen lang:
- So haben wir mit Standardkodierung  $14 * 3 = 42 \text{ Bit}$
- Mit Huffman Kodierung haben wir 31 Bit
- (\*) Reduzierung =  $1 - (\text{Kodiert} / \text{StandardKodierung})$
- Reduzierung =  $1 - (31 / 42) = 1 - 0.738 \approx 0.261$
- Reduzierung =  $0.261 \approx 2.6 / 10 = 26 / 100 = \underline{26\%}$
- *Somit haben wir eine Komprimierung von 26%*
- (\*) Ein Ganzes (1) = 100%, Reduzierung = Differenz zwischen einem Ganzen und dem zweiten Wert

# Übungsblatt 3 - Abgabe bis 6.12.2020 20:00 GTM+1

## Übung 3: Huffman

Einzelaufgabe, 10 Punkte, Abgabe 6.12.2020, 20:00 Uhr in Stud.IP

### 1. Binärzahlen

1 P

- » Was ist die Zahl 17 mit 8 Bit im Binärsystem? 0.2 P
- » Was ist die Zahl 123 im Binärsystem? 0.2 P
- » Was ist die Zahl 010101 im Dezimalsystem? 0.2 P
- » Was ist die höchste Zahl, die man mit 32 Bit darstellen kann? 0.2 P
- » Wie viele Bits braucht es, um jedem Mitgliedsstaat der UN eine eindeutige Nummer zuzuweisen? 0.2 P

### 2. Maximale Rate

1 P

- » Analysiere die Lauflängenkodierung aus der Vorlesung mit der dort angegebenen Steuerbittabelle beim Alphabet  $X = \{\text{Schwarz, Rot, Grün, Blau, Türkis, Magenta, Gelb, Weiß}\}$  bzgl. folgender Frage!
- » Um wie viel % wird die Kodierung eines Text im besten Fall gegenüber der Standardkodierung kürzer?
- » Gib eine begründete Lösung ab!

- *Immer gesamte Rechnung angeben! Endergebnis alleine reicht nicht!*



Das wars mal wieder!

Bis nächste Woche!

