

# Grundlagen der Medieninformatik 1



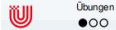
Kodierung  
T08 - 10.11.2021

# Kahoot!

<https://kahoot.it/>

# Übung 2

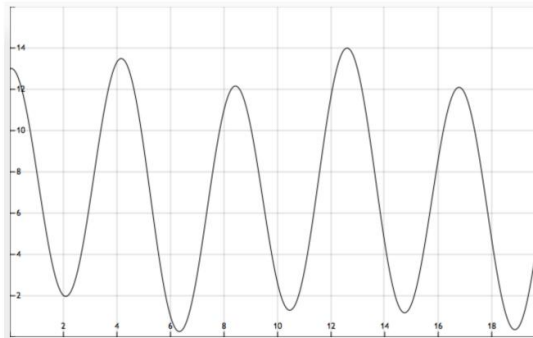
- Abgabe bis 14.11 20:00 Uhr auf StudIP



Übungen  
●○○

## Übung 2: Digitalisierung

Einzelaufgabe, 10 Punkte, Abgabe 14.11.21, 20:00 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

# Abgaben

- Abgaben kommen jeweils in den Ordner unter “Abgabe” welcher dem aktuellen Übungsblatt entspricht
- Folgende Details werden auf jeder Abgabe angegeben:
  - Vor- und Nachname
  - Tutor\*in Name
  - Tutorium #
  - Blatt Nummer
- Dateiname ist immer: mi1\_uebung#\_nachname
- Akzeptierte Formate: PDF und ZIP

Grundlagen der Medieninformatik I  
Tutor\*in: Leonard Haddad  
Tutorium: T08

WiSe 2021/22  
Bearbeiter\*in:

Übungsblatt 1

---

mi1\_uebung1\_haddad.pdf

# Aliasing

- Wir betrachten noch einmal das Thema "Aliasing"
- **Klausurrelevant:** die Bonusaufgabe auf dem vorherigem Tutorium
- Noch eine Aufgabe:
  - "Ein Helikopter hat 5 Flügel im Hauptrotor der sich mit 360 Umdrehungen/min dreht.
  - Bei welcher Samplingfrequenz scheinen die Flügel stillzustehen (die größte Samplingfrequenz ist gesucht)."
- **Lösung:**
  - Der Rotor dreht sich mit 360 Umdrehungen die Minute. Die Rotoren scheinen stehen zu bleiben, wenn zwischen je zwei Samples die Rotoren ihre Stelle so verändern, dass sie jeweils mit dem nächsten Rotor überlappen.
  - Bei 360 U/min sind das insgesamt  $360 * 5 = 1800$  Samples oder 30/s (entspricht dem üblichem Video format von 30 FPS)

# Aliasing - Part 2

- “Bei welcher Samplingfrequenz drehen sie sich nicht mehr vorwärts?”
- Nach dem Nyquist Theorem: zwischen je zwei Samples darf sich der Rotor um maximal die Hälfte der Distanz zwischen je zwei Rotoren bewegen
- Bei 1800 Samples sind das  $1800 * 2 = 3600$  Samples/min oder 60 Samples/s
- Warum? Es geht hierbei um Samples, nach Nyquist:
  - $F_{sampling} \geq 2 * F_{signal}$
  - $F_{signal} = 1800$  (der Rotor überlappt mit dem Nächsten, wir suchen nun die Frequenz bei der er in der Hälfte des Weges zwischen je zwei Rotoren stehen bleibt)
  - Da wir die kleinste Frequenz suchen (“bei welcher”), ist es 60 Samples/s

# Fragenkatalog

- Wichtigste Frage: nutzt ihr den Fragenkatalog überhaupt?
- Wenn nein: solltet ihr, die Fragen kommen in der Klausur in leicht abgeänderter Form dran
- Die MI1 Klausur ist die einfachste die ihr schreiben werdet (aus eigener Erfahrung), es ist sehr einfach diese mit einer 1 zu bestehen wenn ihr den Stoff der Vorlesung durcharbeitet.
- **Nun die Frage: Gibt es noch Fragen zu den Fragen aus dem Fragenkatalog? (Sagt den Satz zehnmal schnell)**

# Kodierung

- Binärdarstellung von Symbolen
- Ein digitales Signal wird durch Symbole dargestellt
- Jede Symbolmenge lässt sich binär kodieren
- $n$  Bits haben  $2^n$  Kombinationen, unzwär die Zahlen  $0..2^n - 1$
- Um auszurechnen wie viele Bits für einen gegebenen Wertebereich ( $0..k-1$ , **insgesamt  $k$  Werte**) gebraucht werden, muss man die Gleichung  $k \geq 2^n$  auflösen. Dieses könnt ihr indem ihr den Logarithmus 2 von  $k$  berechnet und **hochrundet**. E.g.  $\lceil \log_2 k \rceil$



# Kodierung

- Um auszurechnen wie viele Bits für einen gegebenen Wertebereich (0..k-1, **insgesamt k Werte**) gebraucht werden, muss man die Gleichung  $k \geq 2^n$  auflösen. Dieses könnt ihr indem ihr den Logarithmus 2 von  $k$  berechnet und **hochrundet**. E.g.  $\lceil \log_2 k \rceil$
- Habt ihr keinen “ $\log_2$ ” Knopf, so könnt ihr das folgendermaßen mit dem Standardlogarithmus (Basis 10) berechnen:
- $\lceil \log_2 k \rceil = \left\lceil \frac{\log k}{\log 2} \right\rceil$  - Mathematische Regel zum Konvertieren der Logarithmus-Basis

# Kodierungsarten

- Welche Arten von Kodierung haben wir gelernt?
  - Standardkodierung
  - Huffman-Kodierung
  - Lauflängkodierung - Rausgenommen, solltet ihr aber kennen

# Standardkodierung

- Kodierung mit fester Anzahl von Bits
  - Verwende für alle Zeichen gleiche Anzahl von Bits
- Wie viele Bits brauche ich für k Werte?  $k \geq 2^n$  wobei  $n = \min(n_0 \dots n_i)$  - kleinstes n
- Angenommen X das Alphabet.  $X = \{A, B, C, D, E, F, G\}$
- Wir haben  $|X| = 7$  Zeichen und benötigen  $7 \geq 2^n \Rightarrow n = 3$  Bit
- Beispielkodierung:
  - $A \rightarrow 000, B \rightarrow 001, C \rightarrow 010, D \rightarrow 011, E \rightarrow 100, F \rightarrow 101, G \rightarrow 110$

# Standardkodierung

- **Beispiel:** Kodierung von “HALLO WELT” (Leerzeichen ignoriert)
- $X = \{H, A, L, O, W, E, T\}$  -  $X$  ist ein sogenanntes “Alphabet” (eine Menge von Zeichen)
- $|X| = 7 \rightarrow 2^3 = 8 > 7$  - es werden 3 Bit benötigt
- Kodierung z.B.
- $H \rightarrow 000, A \rightarrow 001, L \rightarrow 010, O \rightarrow 011, W \rightarrow 100, E \rightarrow 101, T \rightarrow 110$

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

- “HALLO WELT” kodiert ist also: 000001010010011100101010110

# Standardkodierung

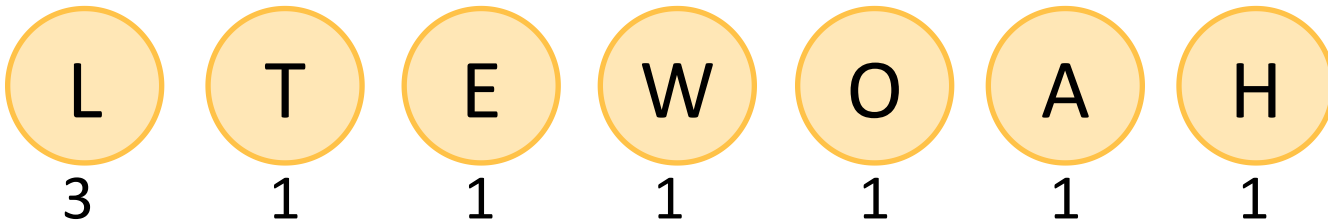
- **Nachteil:**
  - Roher, unkomprimierter Datenstrom
  - → Mehr Speicherverbrauch
- **Lösung:**
  - Komprimierung

# Huffman Kodierung

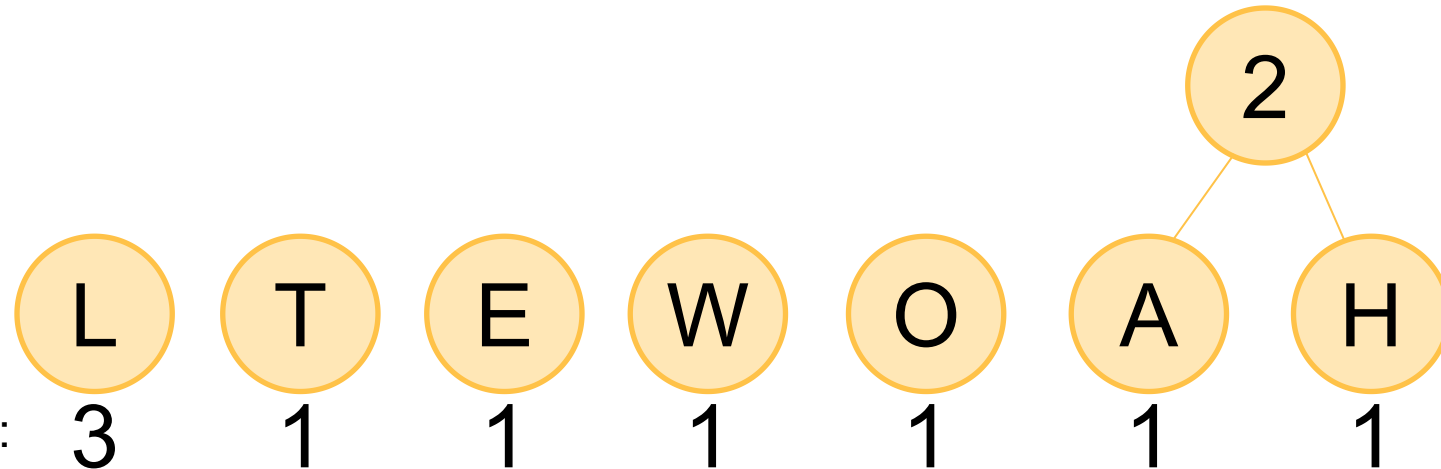
- Idee: Kodiere Zeichen einzeln, aber mit unterschiedlicher Anzahl an Bits, so dass häufige Zeichen kurze Bitfolgen haben
- Vorgehen:
  - Sortiere Zeichen nach Häufigkeit von klein zu groß
  - Verbinde jede 2 kleinst vorkommenden Zeichenknoten
  - Bis alle Knoten als Baum verbunden
- Setze dann noch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

# Huffman Kodierung

- Huffman Kodierung - Beispiel "HALLO WELT"
- 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

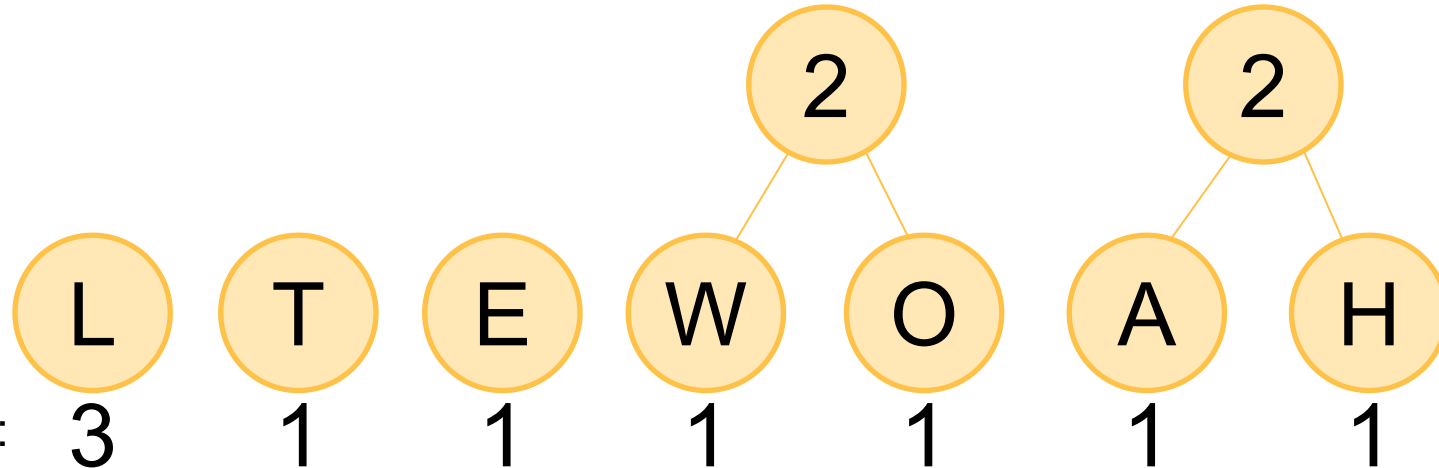


# Huffman Kodierung - Beispiel "HALLO WELT"



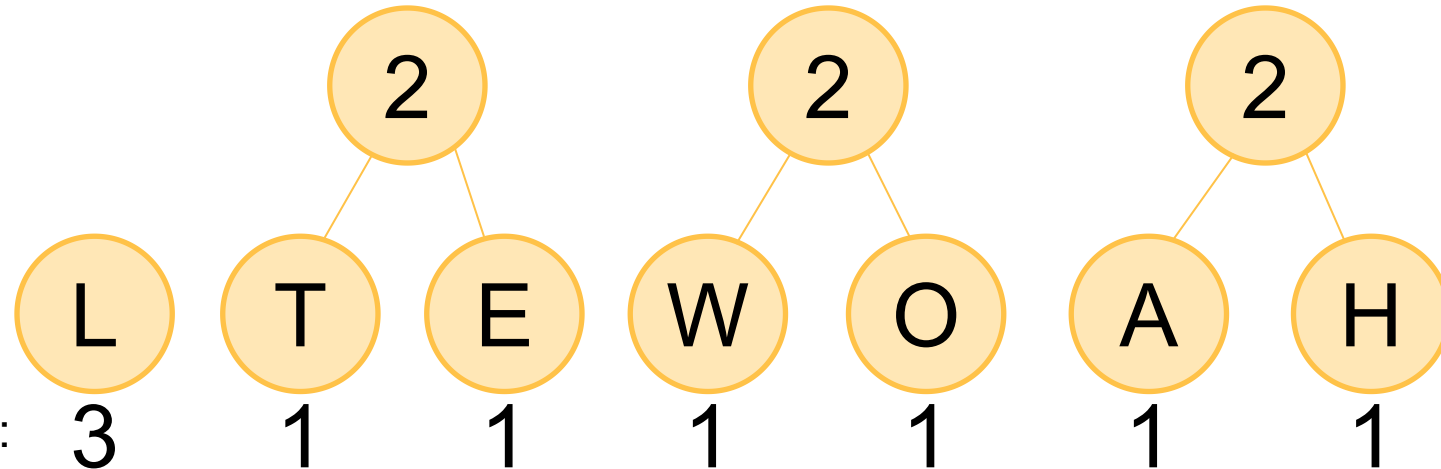


# Huffman Kodierung - Beispiel "HALLO WELT"



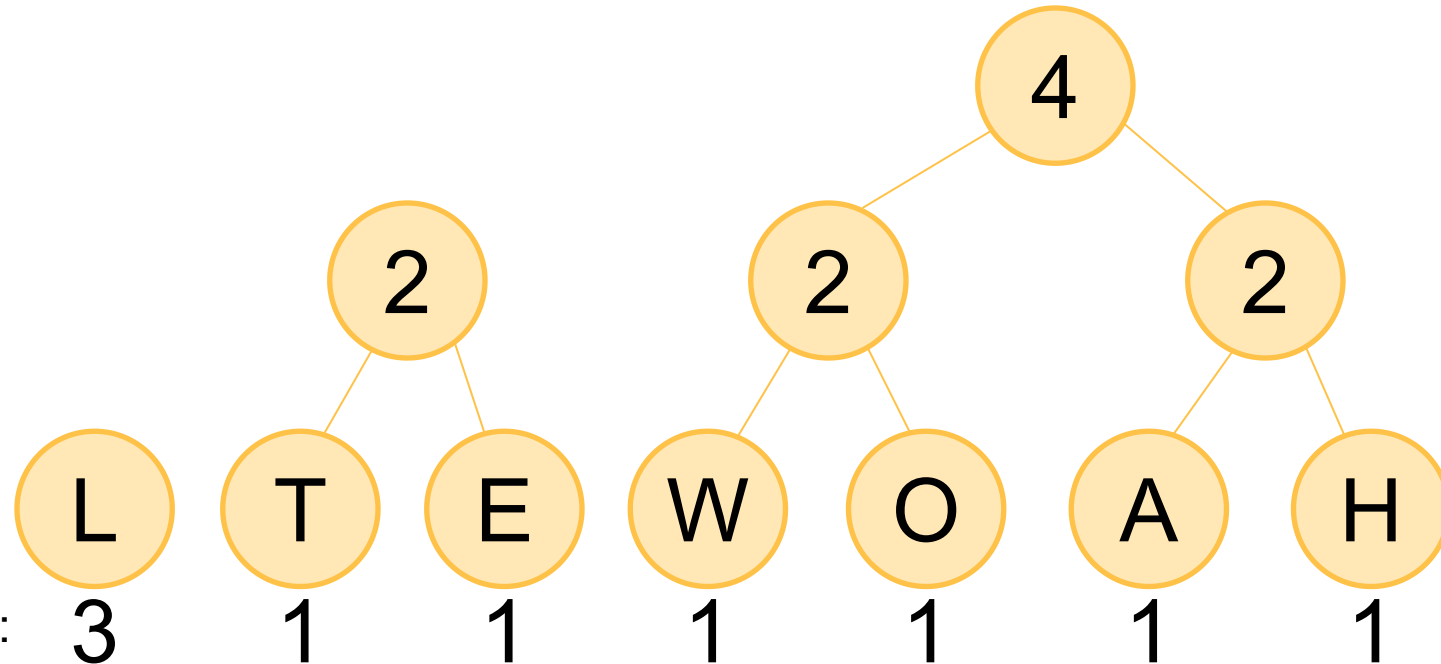
Häufigkeit:

# Huffman Kodierung - Beispiel "HALLO WELT"

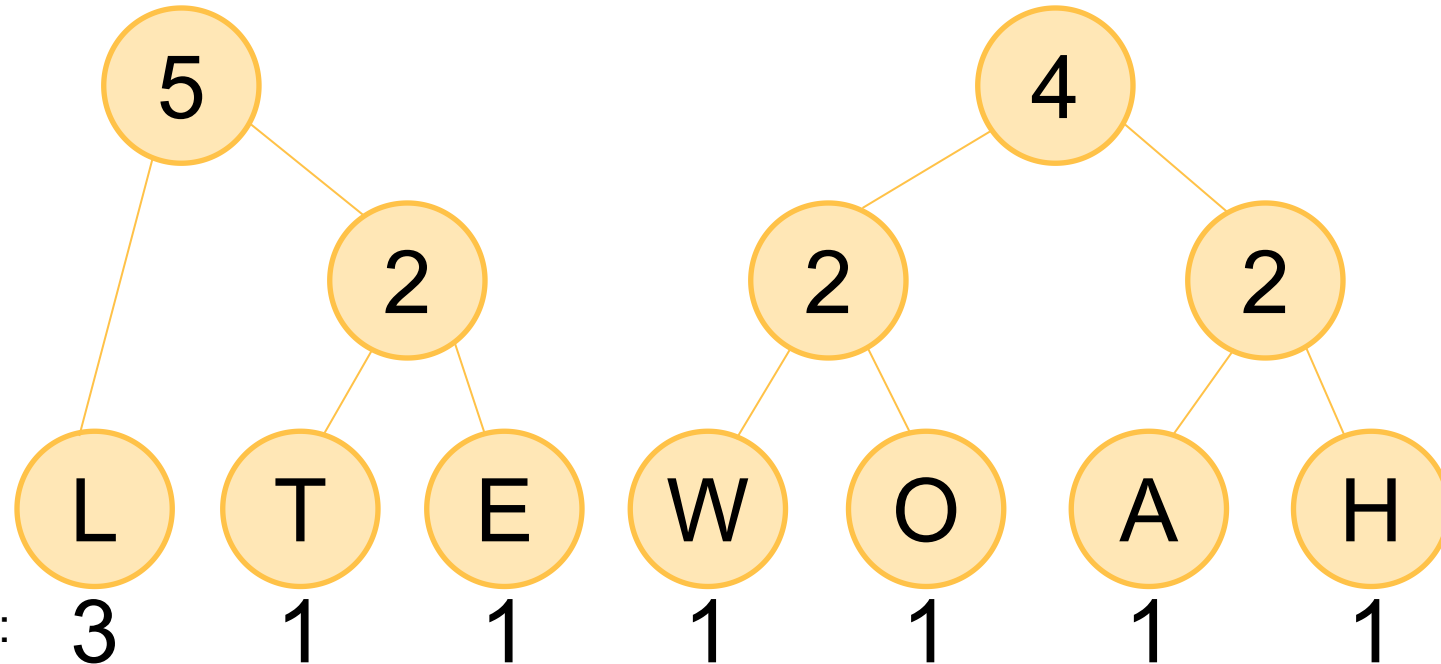


Häufigkeit:

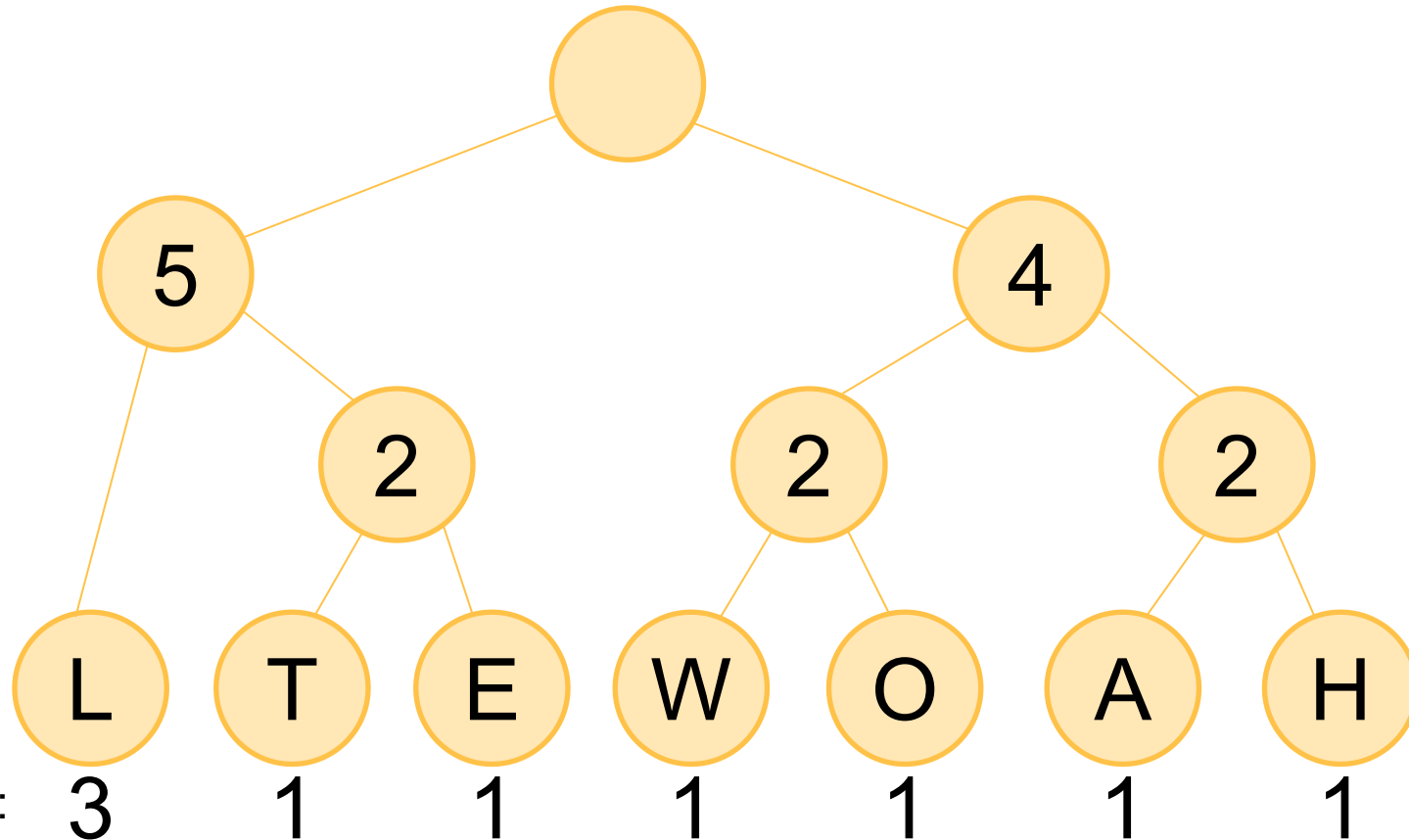
# Huffman Kodierung - Beispiel "HALLO WELT"



# Huffman Kodierung - Beispiel "HALLO WELT"

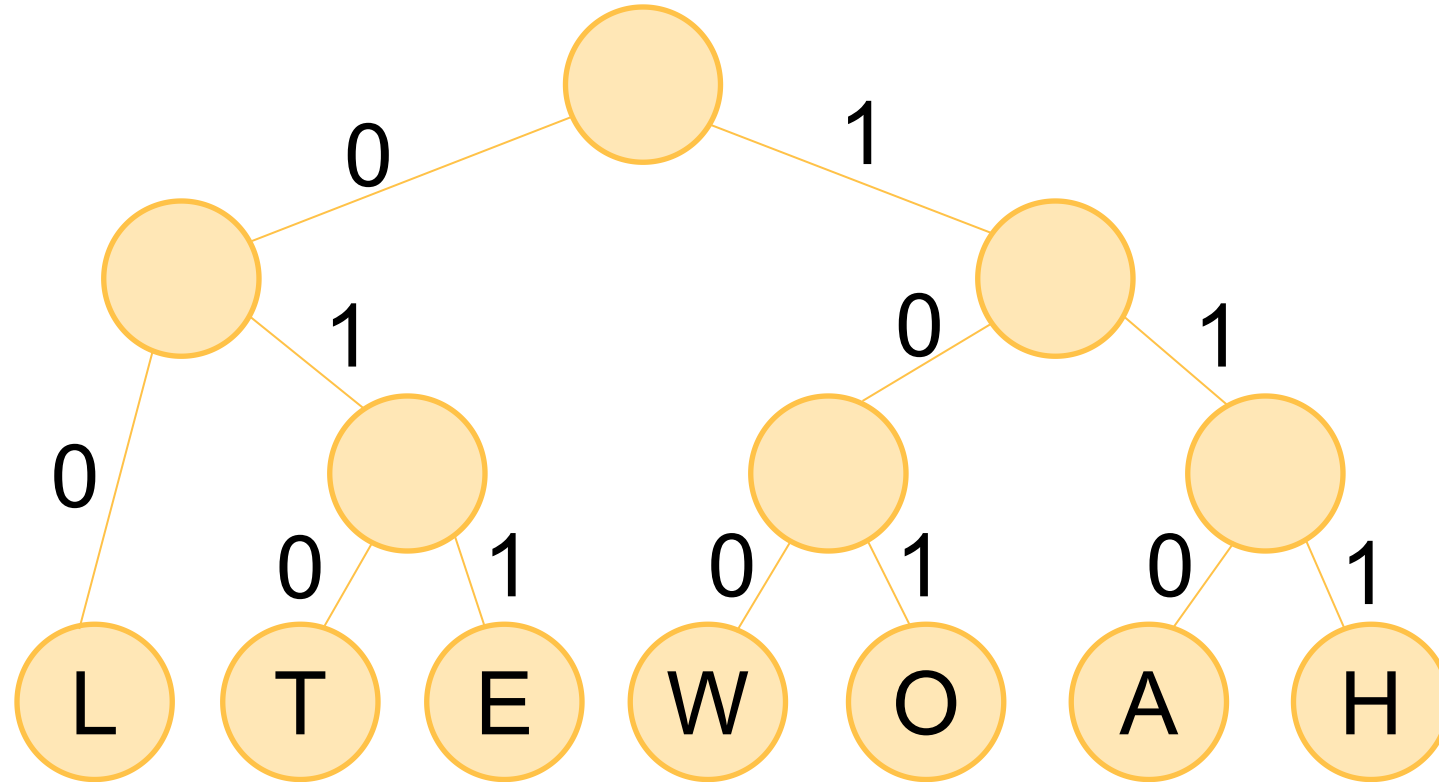


# Huffman Kodierung - Beispiel "HALLO WELT"



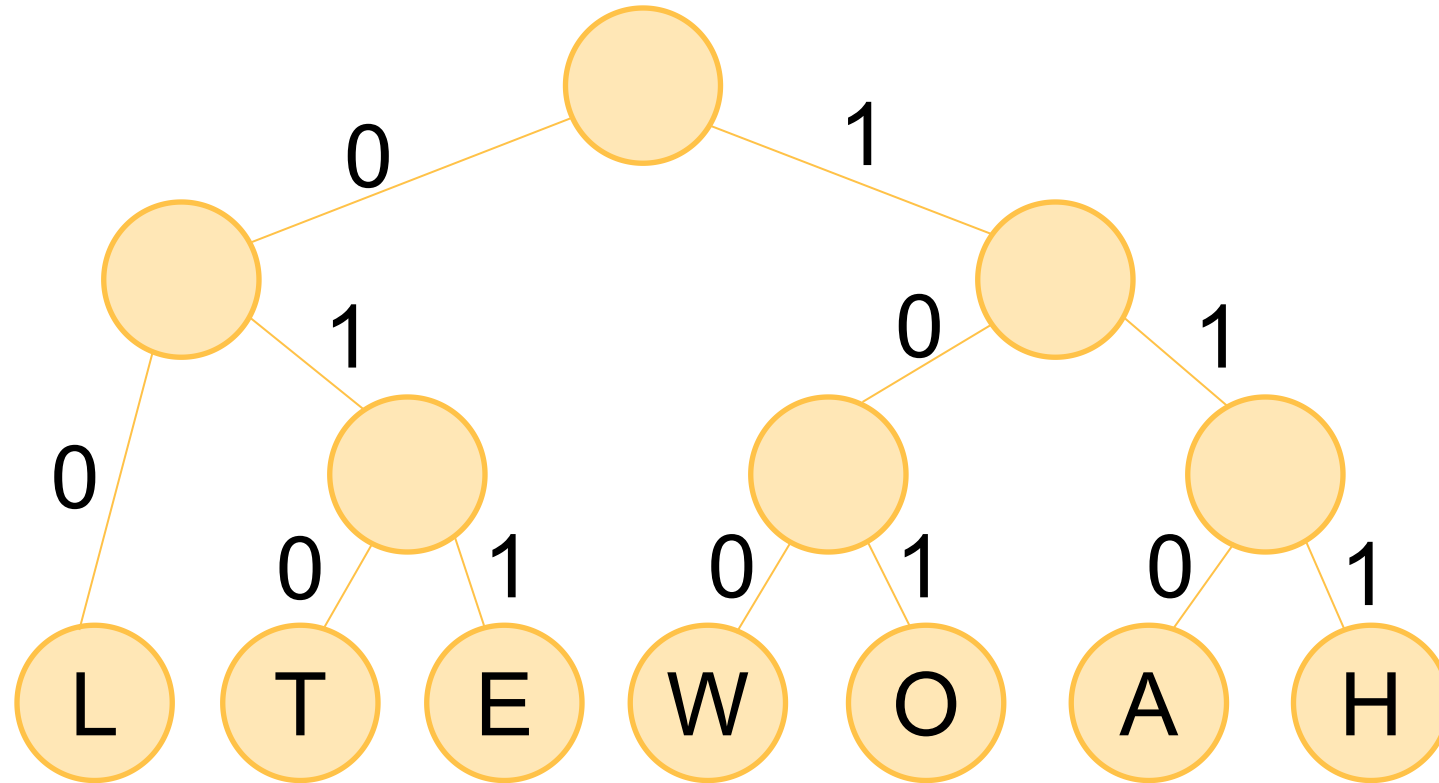
# Huffman Kodierung - Beispiel "HALLO WELT"

- Schreibe 1en Rechts, 0en Links



# Huffman Kodierung - Beispiel "HALLO WELT"

- Dekodiere...



# Huffman Kodierung

## “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” = 111110000010110001100010
- 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

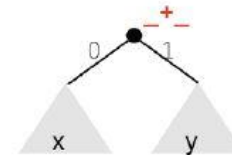
DABADABADU BALU

(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:**

●  
 $x$   
 $n(x)$



# Lösung

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

A  
5

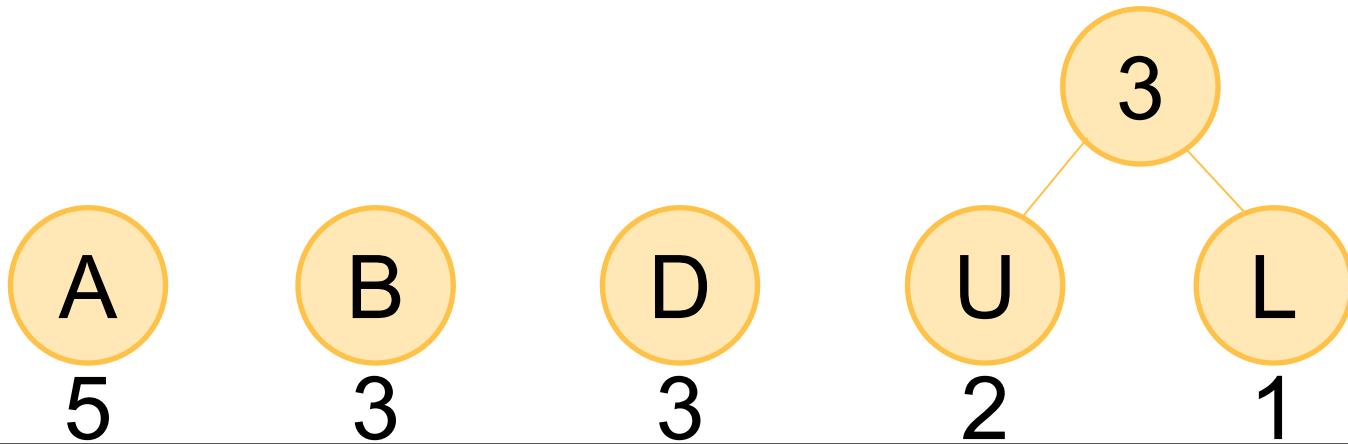
B  
3

D  
3

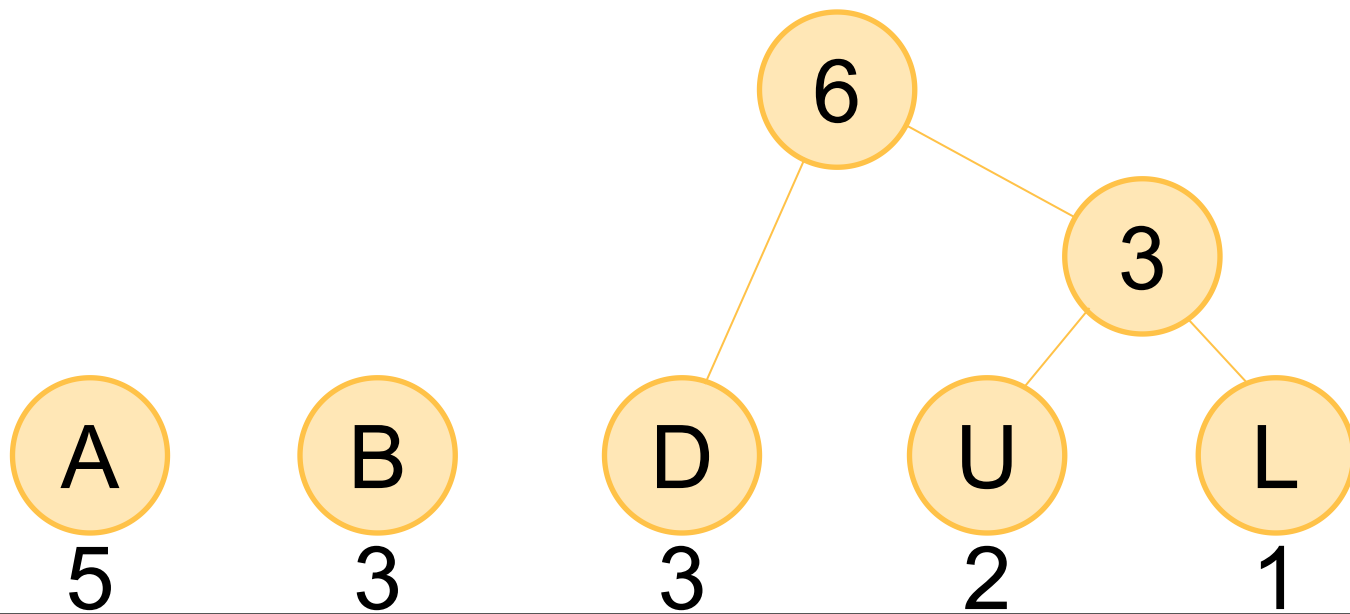
U  
2

L  
1

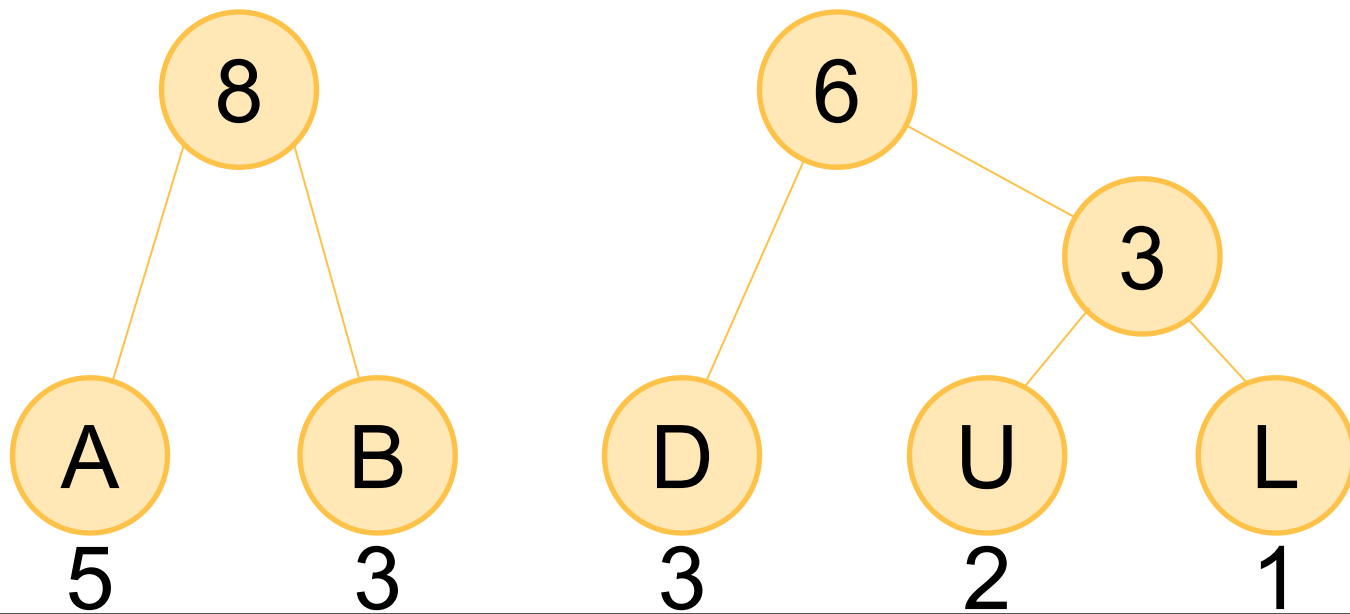
# Lösung



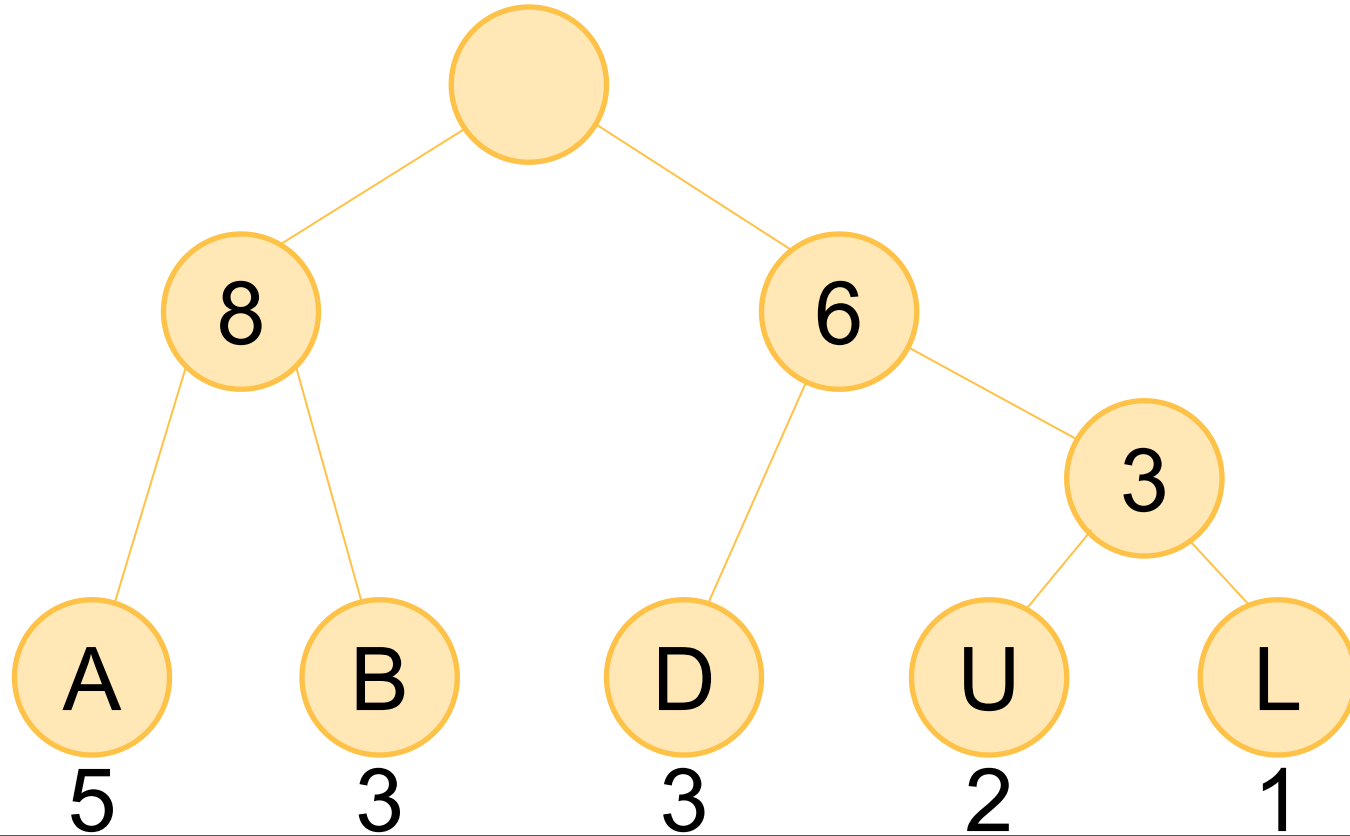
# Lösung



# Lösung

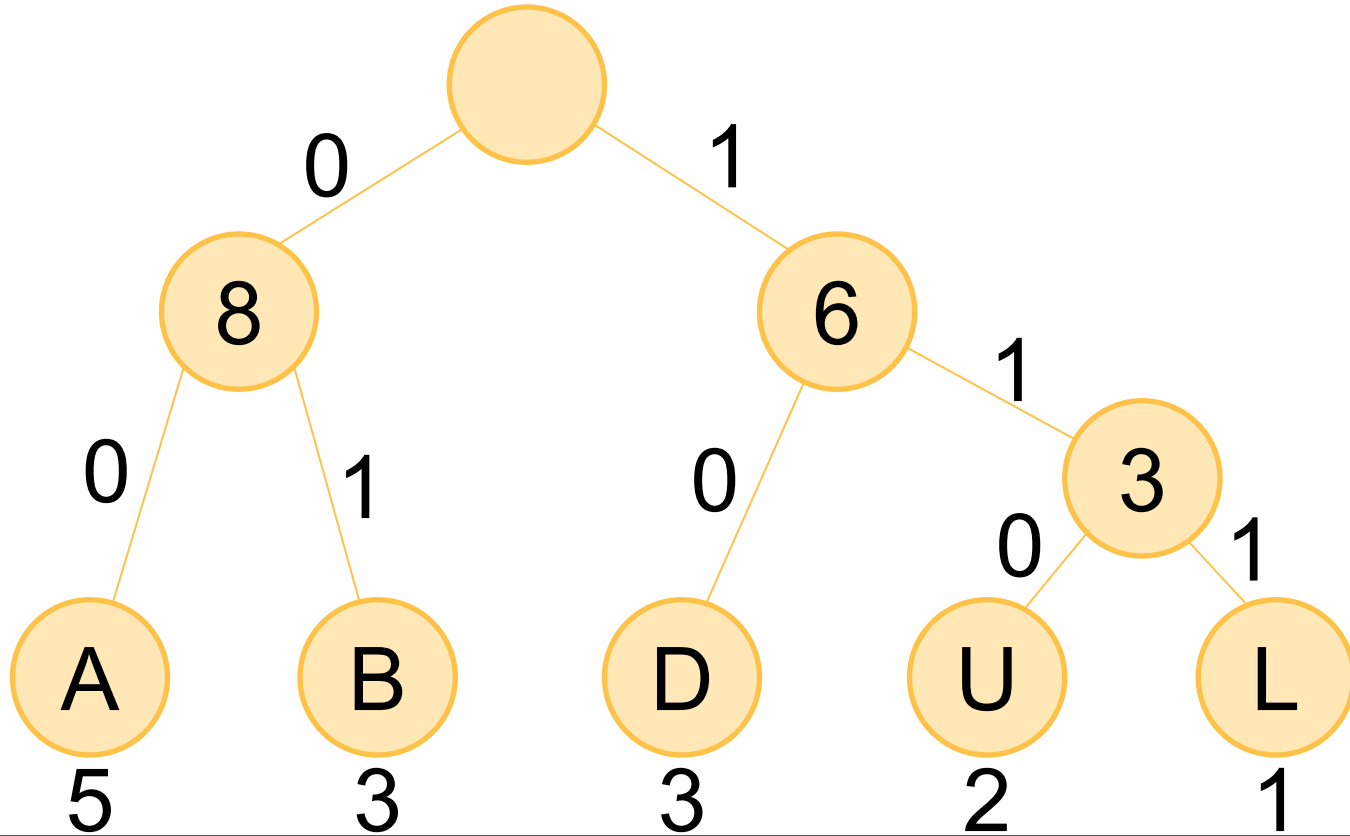


# Lösung



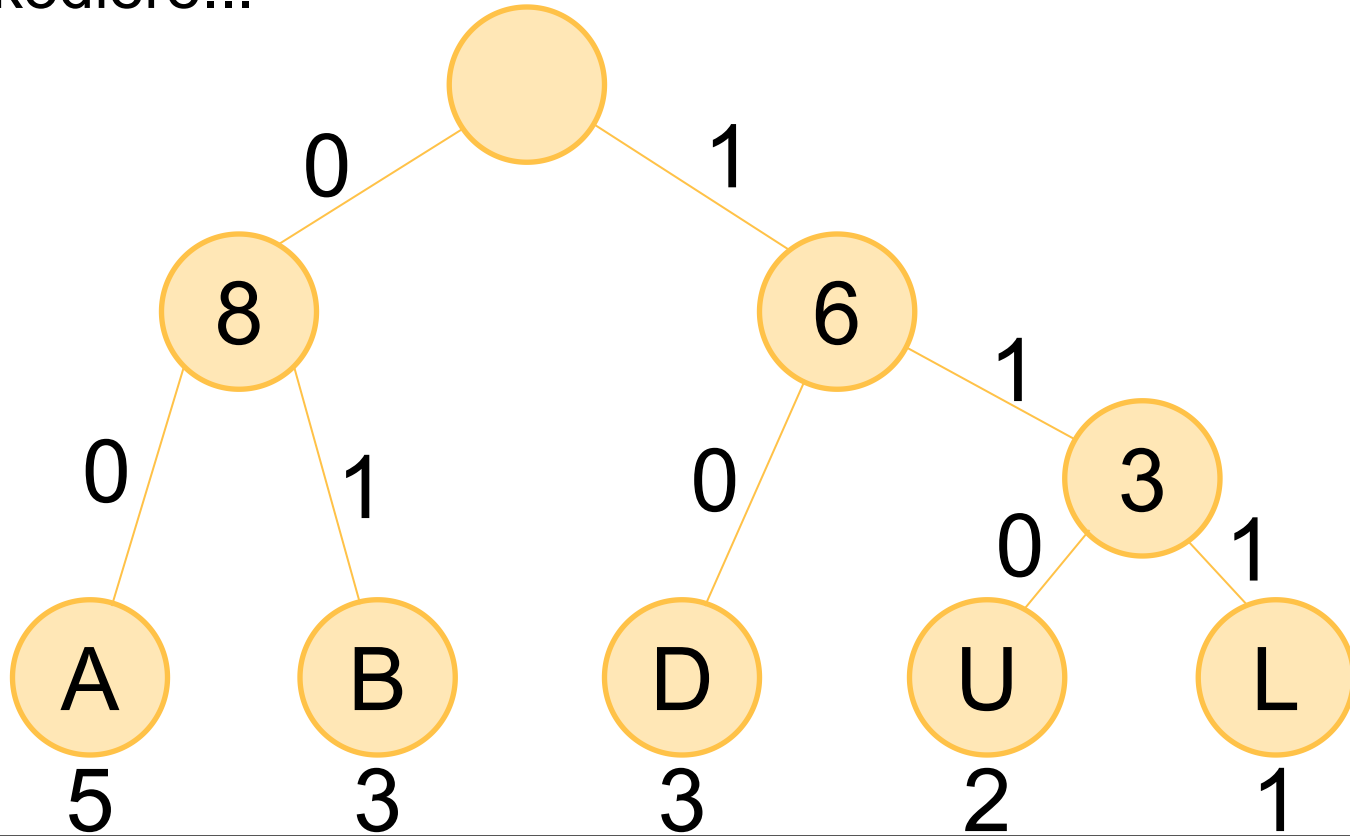
# Lösung

- 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 =  $1 - \frac{31}{42} = 1 - 0.738 \approx 0.261$
  - Reduzierung =  $0.261 \approx \frac{2.6}{10} = \frac{26}{100} = \underline{\underline{26\%}}$
- *Somit haben wir eine Komprimierung von 26%*
- (\*) **Ein Ganzes (1) = 100%, Reduzierung = Differenz zwischen einem Ganzen und dem zweiten Wert**

# Übung 3

- Abgabe bis 21.11 20:00 Uhr auf StudIP



Übungen  
●○○○

## Übung 3: Huffman

Einzelaufgabe, 10 Punkte, Abgabe 21.11.2021, 20:00 Uhr in Stud.IP

### 1. Binärzahlen

2.5 P

- » Was ist die Zahl 37 mit 8 Bit im Binärsystem? 0.3 P
- » Was ist die Zahl 93 im Binärsystem? 0.3 P
- » Was ist die Binärzahl 011101 im Dezimalsystem? 0.3 P
- » Was ist die höchste Zahl, die man mit 24 Bit darstellen kann? 0.3 P
- » Wenn 00 schwarz ist und FF weiss, beides als Hexadezimalzahl, was ist dann ein mittleres Grau als Hexadezimalzahl? 0.3 P
- » Wie viele Bits braucht es, um jedem Mitgliedsstaat der UN eine eindeutige Nummer zuzuweisen? 0.5 P
- » Was ist der rundeste Geburtstag, den Noerdman vermutlich erlebt? 0.5 P
- » Gebe jeweils eine Rechnung oder Begründung an.



# **Das Wars!**

Bis nächste Woche!