

# Huffmanovo kódování

je příkladem bezztrátového kompresního algoritmu. Nejčtenějším znakům přiřazuje kratší kódovací sekvenci, čímž šetří místo. Je detailně popsán např. na [anglické wiki](#), existuje [online applet](#), který provede rozbor na strom libovolné zadané věty.

## Příklad komprese textu

Mějme větu: **TENTO TEXT JE JEN TEST**

V klasickém kódování ASCII zabere 22 B neboli 176 b.

V Huffmanově kódování pouze 60 b tedy necelých 8 B.

## Postup

1. **Zjistíme četnost** jednotlivých znaků:

<b>Znak</b>	T	E	SPC	N	J	O	X	S
<b>Četnost</b>	6	5	4	2	2	1	1	1

2. **Vytvoříme strom** četností: Vezmeme dva znaky s nejnižší četností, vytvoříme z nich nový „složený znak (dvoznak)“, jehož četnost odpovídá součtu četností znaků ve dvojici. To opakujeme tak dlouho, dokud nezbude jen jediný „složený znak“ ze všech znaků obsažených ve zprávě.
3. Zpětně **procházíme vzniklý strom** a na každé úrovni **přidělujeme vždy symboly 1 a 0** tak, že větve/znak s vyšší četností dostane 1 a větve/znak s nižší četností 0.

## Na příkladu

1. Nejnižší četnost mají znaky X,S, proto vznikne dvoznak (X,S) s četností 2:

<b>Znak</b>	T	E	SPC	N	J	(X,S)	O
<b>Četnost</b>	6	5	4	2	2	2	1

2. Další dvojice v pořadí je O a dvoznak (X,S):

<b>Znak</b>	T	E	SPC	((X,S),O)	N	J
<b>Četnost</b>	6	5	4	3	2	2

3. Nyní mají nejnižší četnost znaky N a J, vzniká z nich dvoznak:

<b>Znak</b>	T	E	SPC	(N,J)	((X,S),O)
<b>Četnost</b>	6	5	4	4	3

4. V dalším kroku budeme slučovat trojznak XSO a dvoznak NJ, který bude mít hodnotu 7:

<b>Znak</b>	((N,J),((X,S),O))	T	E	SPC
<b>Četnost</b>	7	6	5	4

5. Dále slučujeme mezeru a E:

<b>Znak</b>	(E,SPC)	((N,J),((X,S),O))	T
<b>Četnost</b>	9	7	6

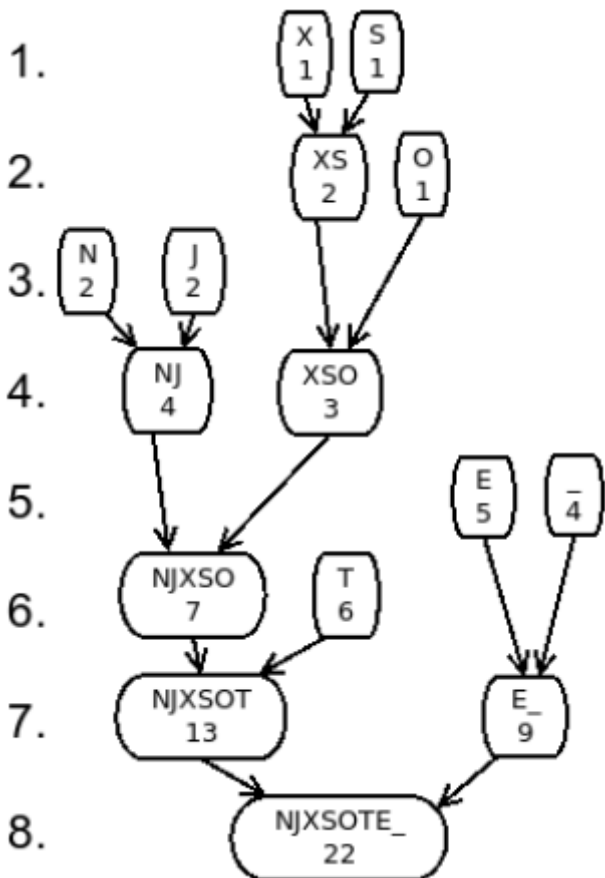
6. V předposledním kroku bude složený znak NJXSO spojen s T:

<b>Znak</b>	( ( (N,J),( X,S),O ),T)	(E,SPC)
<b>Četnost</b>	13	9

7. Při posledním složení už vznikne jediný složený znak, tedy kompletní strom: - V předposledním kroku bude složený znak NJXSO spojen s T:

<b>Znak</b>	( ( ( (N,J),( X,S),O ),T),(E,SPC))
<b>Četnost</b>	22

**Výše uvedený řetězec lze zapsat také formou stromu**



**Při zpětném sestavení přidělujeme 1 a 0**

krok:	1	2	3	4	5	Celkem
<b>T</b>	1	0				<b>10</b>
<b>E</b>	0	1				<b>01</b>
<b>SPC</b>	0	0				<b>00</b>
<b>N</b>	1	1	1	1		<b>1111</b>
<b>J</b>	1	1	1	0		<b>1110</b>
<b>O</b>	1	1	0	0		<b>1100</b>
<b>X</b>	1	1	0	1	0	<b>11010</b>
<b>S</b>	1	1	0	1	1	<b>11011</b>

[online zpracování stromu](#)

## Zakódovaný text

100111111011000010011101010001110010011100111110010011101110

Dekódování probíhá **sekvenčně** náhradou za znak z tabulky, vždy jakmile to nalezená sekvence umožňuje:

10	01	1111	10	1100	00	10	01	11010	10	00	1110	01	00	1110	01	1111	00	10	01	11011	10
T	E	N	T	O		T	E	X	T		J	E		J	E	N		T	E	S	T

From:

<http://wiki.gml.cz/> - GMLWiki

Permanent link:

<http://wiki.gml.cz/informatika:maturita:huffmann>

Last update: **26. 09. 2020, 23.18**

