

# A11 - CAL

Vinicius Gasparini e Lucas Meneghelli

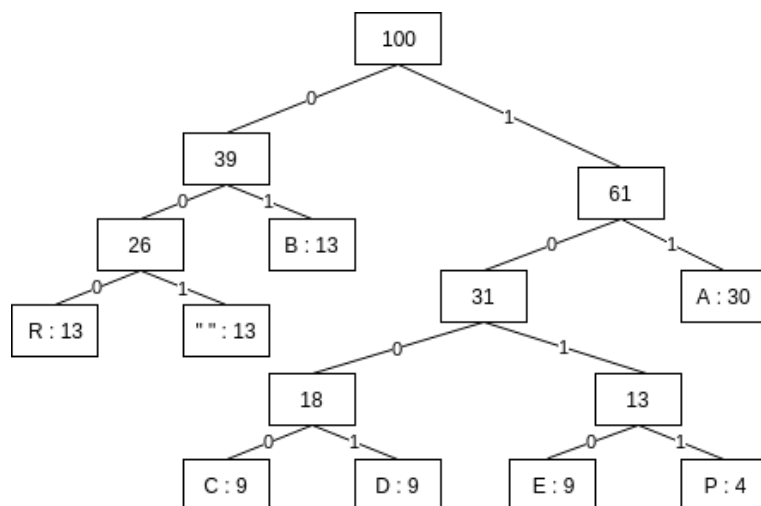
22 de outubro de 2019

## 1 Codificação de Huffman

Foi construído a seguinte tabela relacionando a ocorrência em que cada simbolo aparece na palavra e então medido sua frequência.

	Ocorrência	Frequência	Simples	Huffman
A	7	30	000	11
B	3	13	001	01
" "	3	13	010	001
R	3	13	011	000
C	2	9	100	1000
D	2	9	101	1001
E	2	9	110	1010
P	1	4	111	1011

A árvore usada para construção foi a seguinte



A palavra codificada de maneira fixa é dada por  
000001011000100000101000001011000010111110010101110010100000001011000 = 69 bits

Enquanto a palavra codificada pelo método de Huffman é  
110100011100011100111010001100110111010001100110100011000110100011 = 66 bits

## 2 Puzzle

Para resolver o problema foi utilizado uma otimização sob o algoritmo BFS.

O grafo é modelado de maneira que um estado do jogo representa um vértice. Este vértice então é ligado a estados alcançáveis a partir do jogo atual. O peso da aresta é denotado pela função  $c(x)$  abaixo descrita:

$$c(x) = f(x) + h(x) \text{ onde,}$$

$f(x)$  é a distância do nodo  $x$  até o estado inicial (movimentos até este ponto)  
 $h(x)$  a quantidade de peças que ainda não estão na sua posição final

Tendo em mãos este grafo, é aplicado um BFS priorizando as arestas com menor custo de  $c(x)$ . Caso este caminho não resulte na solução, usamos *backtracking* para voltar a um estado válido.