

Sesión extra

November 1, 2021

Contents

1	Diseñar soluciones	1
2	Árboles de Huffman	1
2.1	Algoritmo	2
2.2	Algún ejemplo	2
2.3	Uso	2
2.4	Como guardar la información	2
2.5	Funciones necesarias	2
2.5.1	Encontrar frecuencias	2
2.5.2	Ordenar	2
2.5.3	Juntar frecuencias en una nueva frecuencia	3
2.5.4	Encontrar nuevos mínimos	3
2.5.5	Recorrer el árbol para generar códigos	3
2.5.6	Codificar	3
2.5.7	Decodificar	3
3	Referencias	3

1 Diseñar soluciones

- Dividir el problema en subproblemas
- Pequeños e independientes

2 Árboles de Huffman

- Códigos perfectos

- Árboles re prefijos
- No muy usados en la práctica

2.1 Algoritmo

1. Ordenas los caracteres por frecuencia
2. Juntar las dos frecuencias más pequeñas para formar una nueva frecuencia
3. Realizar el paso anterior hasta solo tener una raíz
4. Generar códigos recorriendo el árbol de la raíz a. 0 cuando se vaya a la izquierda b. 1 cuando se vaya a la derecha

2.2 Algún ejemplo

2.3 Uso

- Para codificar, concatenar el código de cada caracter
- Para decodificar, usar el árbol para encontrar el prefijo único que corresponde al caracter decodificado

2.4 Como guardar la información

- Entrada y salida
- Frecuencias
- Códigos
- Árbol

2.5 Funciones necesarias

2.5.1 Encontrar frecuencias

```
freqs :: [a] -> [(a, Int)]
```

2.5.2 Ordenar

```
sort :: Ord a => [a] -> [a]
```

2.5.3 Juntar frecuencias en una nueva frecuencia

```
merge :: Tree a -> Tree a -> Tree a
```

2.5.4 Encontrar nuevos mínimos

- Montículos
- Estructura ordenada

2.5.5 Recorrer el árbol para generar códigos

```
getCodes :: Ord a => Tree a -> [(a, [Bit])]
```

2.5.6 Codificar

```
encode :: Ord a => [a] -> (Tree a, [Bit])
```

2.5.7 Decodificar

```
decode :: Ord a => [Bit] -> Tree a -> [a]
```

3 Referencias

- Huffman Encoding · Arcane Algorithm Archive