# Greedy: Huffman Coding

Wednesday, 6 September 2023 11:23 AM

## GREEDY ALGORITHMS:

**Example 1:** Given an undirected graph $G = (V, E)$ find a matching $M \subseteq E$ of large size (i.e, a set of edges s.t. no 2 edges have the same vtx.)

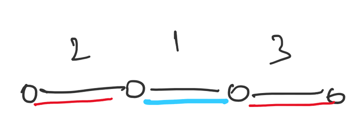
Greedy algo:

$M \leftarrow \phi$

For some arbitrary ordering on the edges $e_1 \dots e_m$

If $M \cup e_i$ is a matching, add $e_i$ to $M$.

Example:




**Theorem:** Greedy algo produces a matching of size at least $\frac{1}{2}$ the size of the maximum matching.

(Thus greedy algorithm obtains a $\frac{1}{2}$-approximation to the maximum matching)

Proof left as an exercise

**Example 2:** Dijkstra's algorithm for finding a shortest path in a graph

(will do this algorithm later in class)

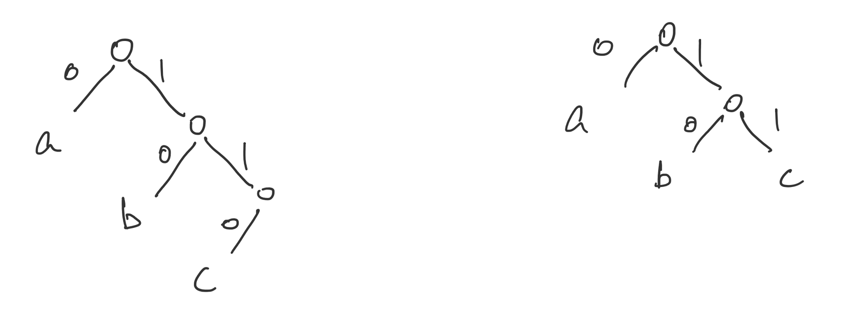
**Example 3:** Huffman Coding.

**Given:** Symbols $a_1, a_2, \dots, a_n$

A prefix-free coding of $a_1, a_2, \dots, a_n$ assigns a binary string to each symbol s.t. no string is a prefix of the other.

| | | | |
|---|---|---|---|
| a | 0 | 0 | 111 |
| b | 01 | 10 | 011 |
| c | 011 | 110 | 001 |
| | ✗ | ✓ | ✓ |

Point is, given any sequence of bits, receiver should know exactly if a symbol is received or not.




Assignment of binary strings can also be seen as placing the symbols in a binary tree.

Coding is prefix-free if all symbols are at leaves of the tree

Additionally given frequencies $f_1, \dots, f_n$ for the symbols.

Then for a tree $T$, $\text{cost}(T) = \sum_{i=1}^{n} f_i \, ht_T(a_i)$

Find a binary tree of minimum cost, where all symbols are at the leaves.

**Example:**

| | a | b | c | d | e |
|---|---|---|---|---|---|
| f: | 5 | 9 | 12 | 13 | 16 |

Following algorithm:

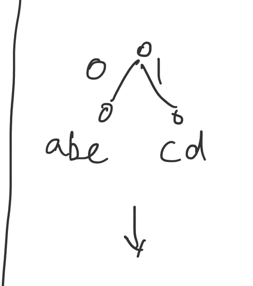
ab c d e → ab cd e

14 12 13 16 → 14 25 16

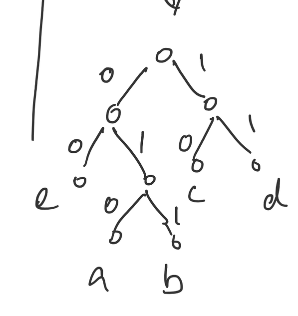
↓

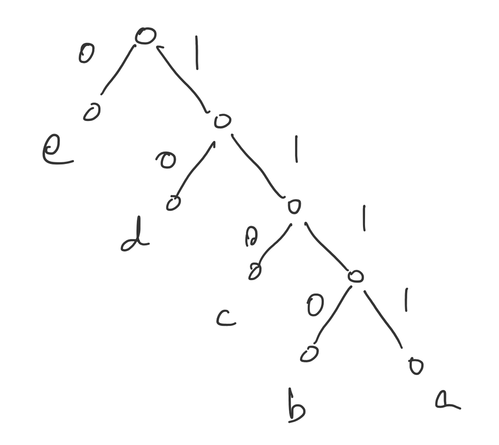
abe cd

30 25




↓




Code:

| | |
|---|---|
| a | 1111 |
| b | 1110 |
| c | 110 |
| d | 10 |
| e | 0 |

cost of this encoding / tree is 134

(Goal: minimize length of message, where each symbol occurs w/ given frequency)

**Claim 1:** The least frequency element must be at max distance from the root

(prove yourself)

**Claim 2:** In every optimal tree, every internal node has 2 children (i.e., the binary tree is complete)

(trivial)

**Claim 3:** There exists an optimal tree where the two least-frequency elements are siblings (and are hence at max distance from the root)

(prove yourself)

**Lemma:** Given symbols $a_1, \dots, a_n$ w/ frequency $f_1 \dots f_n$. Suppose $a_{n-1}, a_n$ are the least-frequency elements. Consider the problem w/ symbols $a_1, a_2, \dots, a_{n-2}, b_{n-1}$ w/ frequencies $f_1, f_2, \dots, f_{n-2}, (f_{n-1} + f_n)$. Let $T'$ be an optimal tree for the smaller problem. Then an optimal tree for the original problem is obtained by replacing $b_{n-1}$ in $T'$ by an internal w/ children $a_{n-1}, a_n$

**Proof:** Let tree $T$ be obtained from $T'$ by the procedure given. $\text{cost}(T) = \text{cost}(T') + f_{n-1} + f_n$

Suppose $\hat{T}$ is an optimal solution for the original problem w/ smaller cost where $a_{n-1}, a_n$ are siblings. Consider $\hat{T}'$ obtained by replacing $a_{n-1}, a_n$ & their parent by node $b_{n-1}$.

$$\begin{aligned} \text{cost}(\hat{T}') &= \text{cost}(\hat{T}) - f_{n-1} - f_n \\ &< \text{cost}(T) - f_{n-1} - f_n \\ &= \text{cost}(T') \end{aligned}$$

which is a contradiction, since we assumed $T'$ was an optimal tree for the modified problem.

The algo:

```
HCTree(f1, f2, ..., fn) {
    Let i, j be elts. of min-frequency
    T' ← HCTree(f1, ..., f(i-1), f(i+1), ..., f(j-1), f(j+1), ...,
                fn, fi + fj)
    T ← replacing elt in T' w/ freq. fi + fj w/
        an internal & 2 children w/ frequency
        fi, fj
    return T
}
```