CSL 356: Analysis and Design of Algorithms

Ragesh Jaiswal

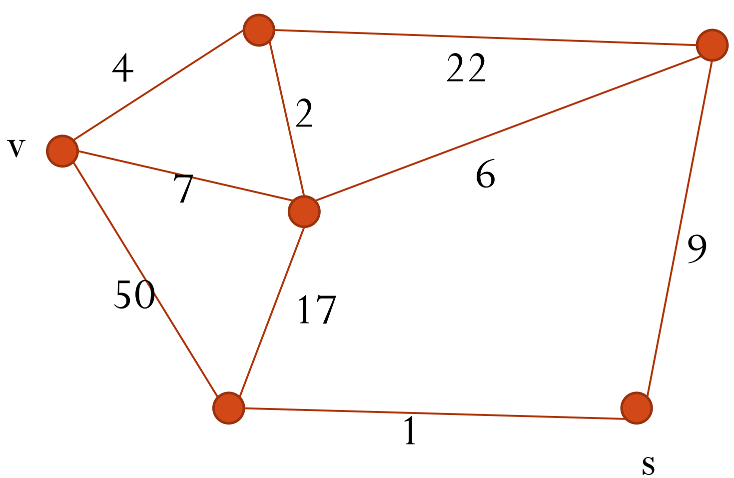
CSE, IIT Delhi

Greedy Algorithms: Examples

Shortest Path: Dijkstra's Algorithm

Greedy Algorithms: Examples

- Let $G = (V, E)$ be a weighted directed graph. Given a path in G, the length of a path is defined to be the sum of lengths of the edges in the path.
- The shortest path from u to v is the path with minimum length.
- <u>Problem</u>: Given a weighted directed graph with positive edge weights and a source vertex s find the shortest path from s to all other vertices in the graph.



Greedy Algorithms: Examples

- Problem: Given a weighted directed graph with positive edge weights and a source vertex s find the shortest path from s to all other vertices in the graph.
- Claim: Shortest path is a *simple path*.

Greedy Algorithms: Examples

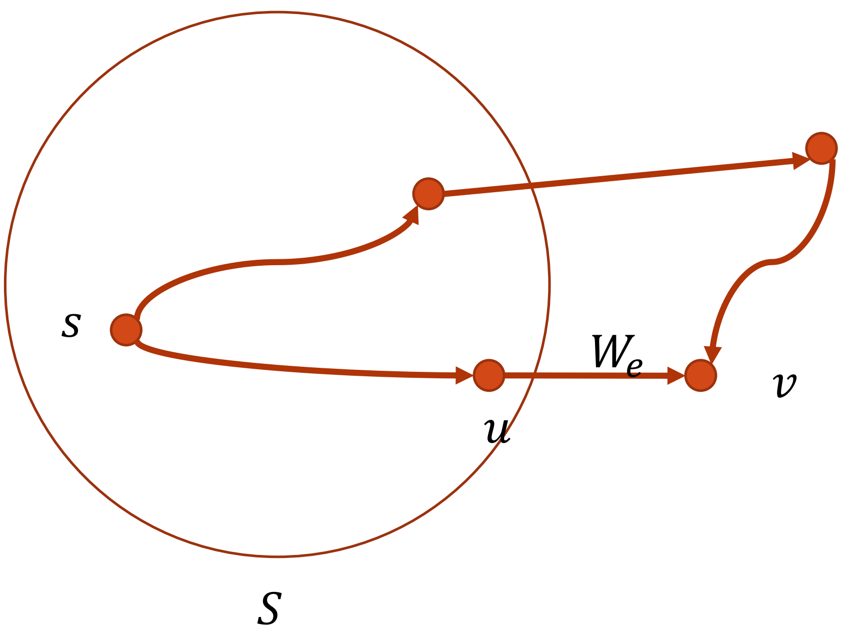
- Problem: Given a weighted directed graph with positive edge weights and a source vertex s find the shortest path from s to all other vertices in the graph.
- Claim: Shortest path is a *simple path*.
- Claim: Let S be a subset of vertices containing s such that we know the shortest path length $l(s,u)$ from s to any vertex in $u \in S$. Let $e = (u,v)$ be an edge such that
  - u is in S, v is in $V \setminus S$,
  - $l(s,u) + W_e$ is the least among all such cut edges.

  Then $l(s,v) = l(s,u) + W_e$

Greedy Algorithms: Examples

- <u>Claim</u>: Let S be a subset of vertices containing s such that we know the shortest path length $l(s,u)$ from s to any vertex in $\mathrm{u} \in S$. Let $e=(u,v)$ be an edge such that
  - u is in S, v is in $V \backslash \mathrm{S}$,
  - $l(s,u) + We$ is the least among all such cut edges.

  Then $l(s,v) = l(s,u) + W_e$



Greedy Algorithms: Example

- Dijkstra's Algorithm:

Dijkstra's Algorithm(G, s)
- $S = \{s\}$
- $d(s) = 0$
- While S does not contain all vertices in G
  - Let $e = (u, v)$ be a cut edge (across S, S') with minimum value of $(d(u) + W_e)$
  - $d(v) = d(u) + W_e$
  - $S = S \cup \{v\}$

- What is the running time?

Greedy Algorithms: Huffman Coding

Greedy Algorithms: Huffman Coding

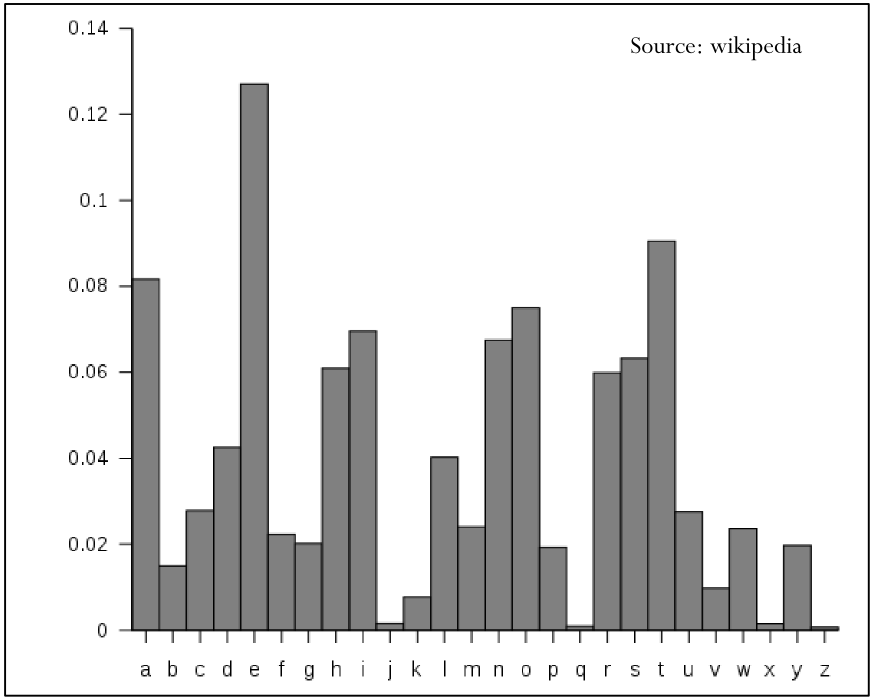
- **A** wants to send an email to **B** but wants to minimize the amount of communication (number of bits communicated).
- How do you encode an email into bits?
  - ASCII: (8 bits per character)
    - Is this the best way to encode the email given that the goal is to minimize the communication?

Greedy Algorithms: Huffman Coding

- **A** wants to send an email to **B** but wants to minimize the amount of communication (number of bits communicated).
- How do you encode an email into bits?
  - ASCII: (8 bits per character)
    - Is this the best way to encode the email given that the goal is to minimize the communication?
    - Different alphabets have different frequency of occurrence in a standard English document.

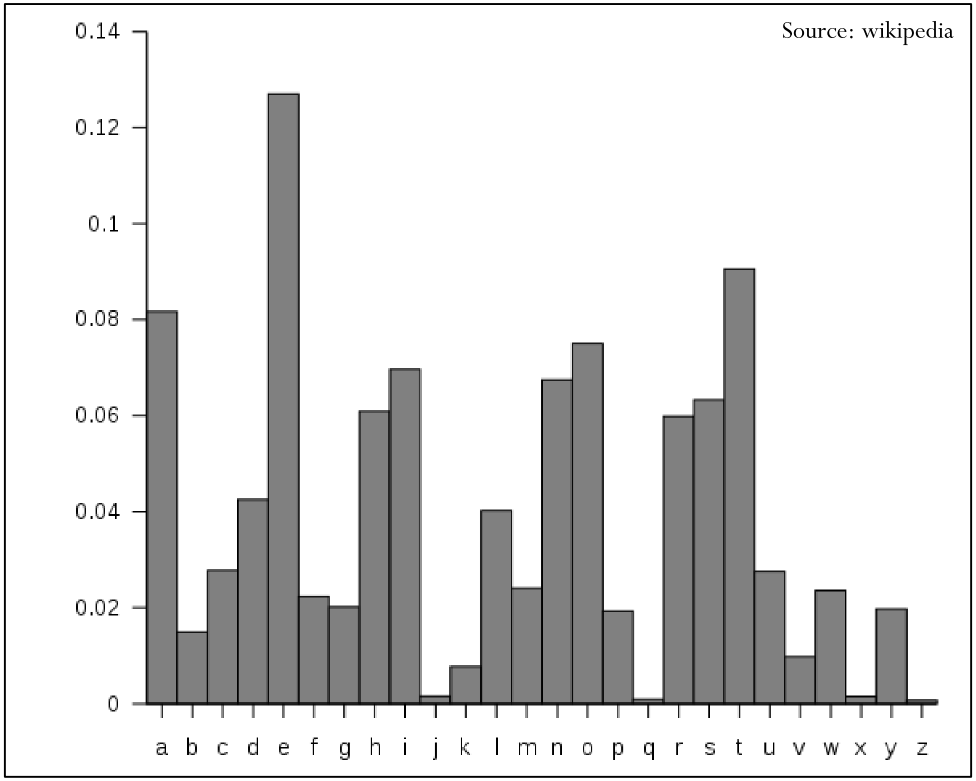
Greedy Algorithms: Huffman Coding

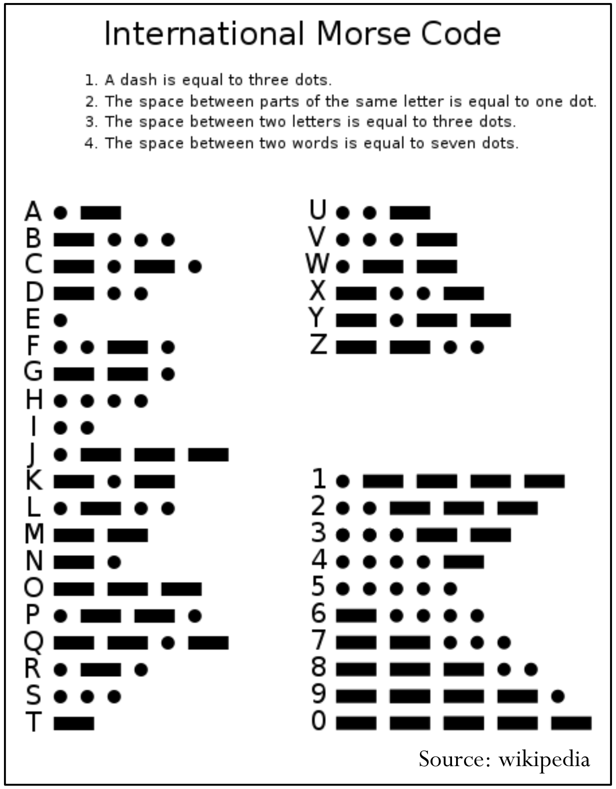
- **A** wants to send an email to **B** but wants to minimize the amount of communication (number of bits communicated).
- How do you encode an email into bits?
  - ASCII: (8 bits per character)
    - Is this the best way to encode the email given that the goal is to minimize the communication?
    - Different alphabets have different frequency of occurrence in a standard English document.



Greedy Algorithms: Huffman Coding

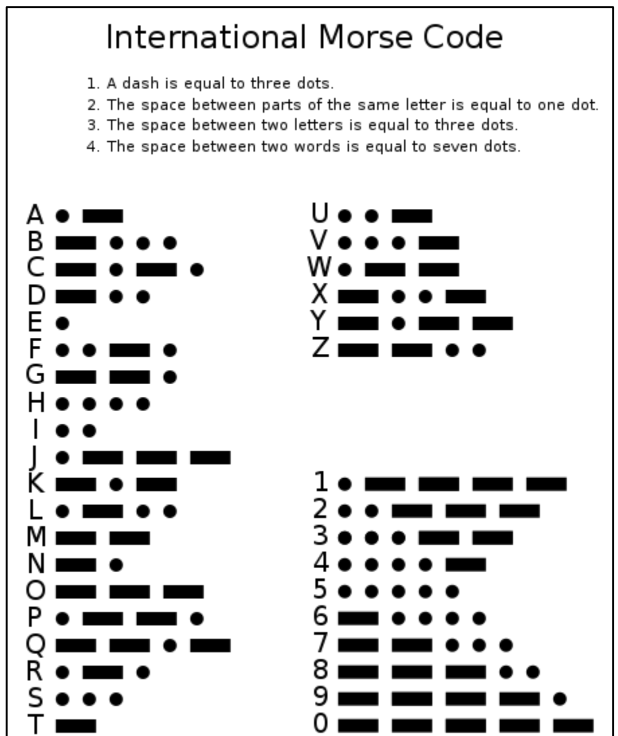
- The encoding of "e" should be shorter than the encoding of "x".
- In fact *Morse code* was designed with this in mind.





Greedy Algorithms: Huffman Coding

- Suppose you receive the following Morse code from your friend:
  - ● ● ● ▬▬▬
  - What is the message?



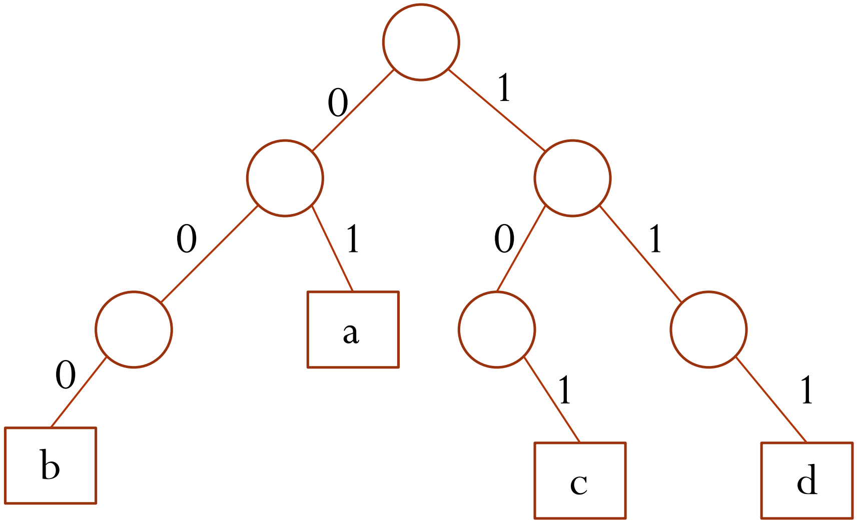
Source: wikipedia

Greedy Algorithms: Huffman Coding

- Prefix-free encoding: An encoding f is called prefix-free if for any pair of alphabets (a_1, a_2), $f(a_1)$ is not a prefix of $f(a_2)$.
- Morse code is clearly not prefix-free.
- Consider a *binary tree* with 26 leaves and associate each alphabet with a leaf in this tree.
  - Binary Tree: A rooted tree where each non-leaf node has at most two children.
- Label an edge 0 if this edge connects the parent to its left child and 1 otherwise.
- $f(x) =$ The label of edges connecting the root with x.

Greedy Algorithms: Huffman Coding

- Consider a *binary tree* with 26 leaves and associate each alphabet with a leaf in this tree.
- Label an edge 0 if this edge connects the parent to its left child and 1 otherwise.
- $f(x) =$ The label of edges connecting the root with x.

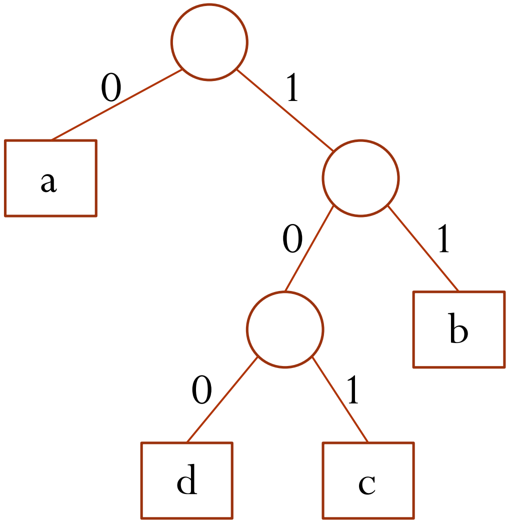


Simple example with 4 alphabets

- $f(a) = 01$
- $f(b) = 000$
- $f(c) = 101$
- $f(d) = 111$
- Is f prefix-free?

Greedy Algorithms: Huffman Coding

- Suppose you are given a prefix-free encoding g.
- Can you construct a binary tree with 26 leaves, associate each leaf with an alphabet, and label the edges as defined previously such that the for any alphabet, the label of edges connecting the root with $x = g(x)$?



Simple example with 4 alphabets

- $g(a) = 0$
- $g(b) = 11$
- $g(c) = 101$
- $g(d) = 100$

End