Why?

- Saves space.
- Reduces I/O.
- Reduces communication load.
- Cryptography.

} Saves Time

Data Compression

DATA → Compression Algorithm → DATA

DATA → Decompression Algorithm → DATA

Example: FAX

- Some fax machines use a version of Run-Length coding as a part of the compression procedure.
- Each row is encoded by a list of integers telling the number of white and black pixels in each group. e.g. 1,8,17,3,...
- Each row starts with a dummy white pixel.
- All rows have the same length.

How?

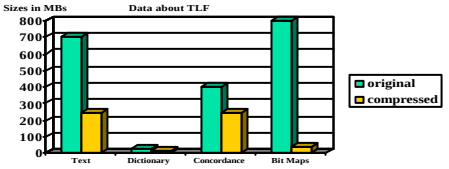
- Data Redundancy
  - Example:
    In an English text file, ftp sends just 7 bits since 8th bit is always 0.
  - Yet another example:
    A string with many 0s and very few 1s - 1 is an innovation. 0 doesn't tell much.

Classification I

- Statistic codes
  - Code length depends on probability of item.
- Dictionary codes
  - The compressed file consists of pointers to a collection of strings.

What?

- TRESOR DE LA LANGUE FRANCAISE (TLF) - 120,000,000 words.
- RESPONSA RETRIEVAL PROJECT (RRP) - 60,000,000 words.



Classification III

- Reversable Codes
  - No loss of data
- Lossy Methods
  - Some data may lost.
  - Very common in use for audio, images, video etc.

Classification II

- Static codes
  - Two passes on data.
  - One for gathering information and one for compressing.
- Adaptive codes
  - One pass on data for both gathering information and compressing.
  - Has ability to "forget".

The prefix property

- No codeword is the prefix of any other
- Example:
  - A 000
  - B 001
  - C 0100
  - D 0101
  - E 011
  - F 10
  - G 11

10011010000000111
F E C A B G

Unique Decipherability

- Fixed length code can be easily broken into codewords.
- How can we split a variable length code?

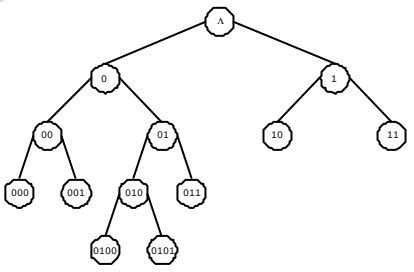
A 0
B 10
C 11
D 1

110 can be CA or DB .

Complete prefix codes

- A prefix code is complete if it corresponds to a complete binary tree.
- A prefix code is complete if an insertion of another codeword will make it not Uniquely Decipherable.

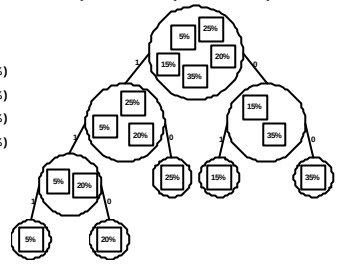
Tree of a Prefix code



Example

- **A - 5%, B - 15%, C - 20%, D - 25%, E - 35%.**

```
111 A(5%)
 01 B(15%)
110 C(20%)
 10 D(25%)
 00 E(35%)
```



Shannon-Fano coding

- Divide the set of symbols into two equal or almost equal subsets based on the probability of occurrence of characters in each subset.
- One set is assigned 0.
- The other set is assigned 1.
- Repeat the procedure until all subsets have a single element.

Shannon-Fano is not optimal

- This tree will give better compression:
- Shannon-Fano tree:

  3*0.05+2*0.15+3*0.20
  +2*0.25+2*0.35=**2.25**
- This tree is better:

  3*0.05+3*0.15+2*0.20
  +2*0.25+2*0.35=**2.2**