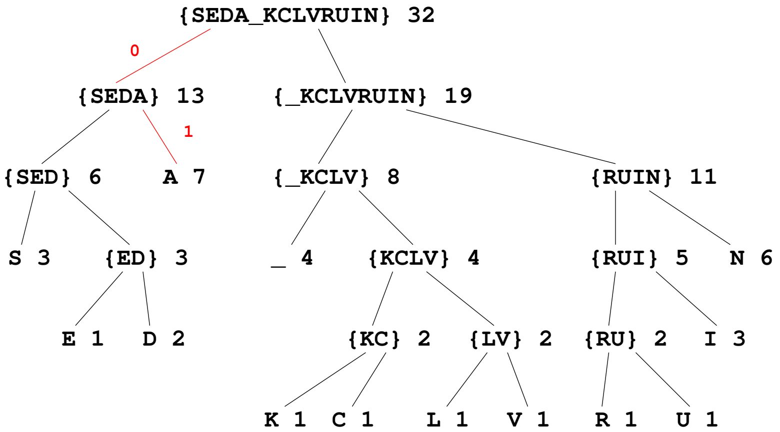
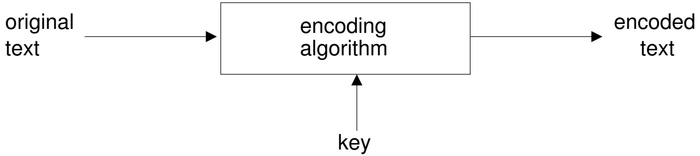
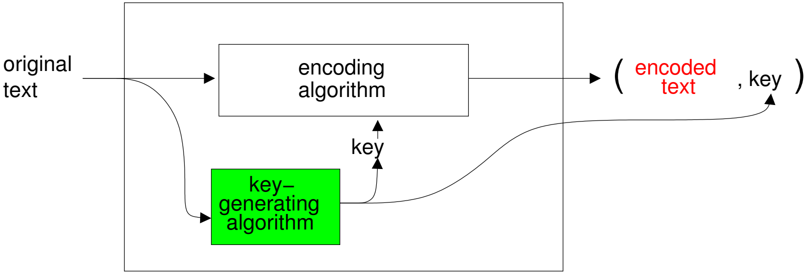
From DMFP:



01	111	111	1101	10100	01
A	N	N	I	K	A


original text
encoding algorithm
encoded text
key


original text
encoding algorithm
(encoded text , key)
key
key–generating algorithm

Lemma 16.2, restated from CLRS pg 433:
Let x and y be characters in n alphabet with the lowest frequencies in the original text. Then there exists an optimal prefix code for the alphabet that is optimal for the original text in which which the encodings of x and y have the greatest length.
Proof sketch. Let T be an optimal tree. Let a and b be characters represented by sibling leaves of maximal depth. By how x, y, a, and b are chosen,

$$
\begin{aligned}
x.freq &\leq a.freq \\
y.freq &\leq b.freq
\end{aligned}
$$

Let T'' be the prefix code like T except with x and a switch, y and b switched. Then

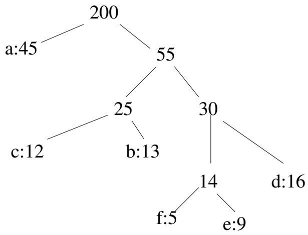
$$
\begin{aligned}
B(T) - B(T'') &= \sum_{c \in C} c.freq \cdot d_T(c) - \sum_{c \in C} c.freq \cdot d_{T''}(c) \\
&= (a.freq - x.freq)(d_T(a) - d_T(x)) \\
&\quad +(b.freq - y.freq)(d_T(b) - d_T(y)) \\
&\geq 0 \qquad \square
\end{aligned}
$$

Lemma 16.3, summarized from CLRS pg 435:
Optimal trees have subtrees that are optimal for their corresponding subproblem.

Theorem 16.4, restated from CLRS pg 435:
Huffman trees are prefix codes that are optimal for the given original text.
Proof sketch. Let C be the alphabet of the text, augmented with character frequencies. By induction on the structure of the tree.
Base case. Suppose C has only on character. Then there is only one possible tree for that alphabet, so it must be optimal.
Inductive case. Suppose C has more than one chacter, and let x and y be the the least frequent characters. Let C' be the alphabet like C but with x and y replaced with pseudo-character z. By structural induction, the Huffman encoding produces a tree that is optimal for C'. By Lemma 16.3, we can replace leaf z in the optimal tree for C' with a parent of siblings x and y to make a tree optimal for C. □

Solution to 16.3-2. Suppose T is a non-full prefix code binary tree. Let x be a node with one child. Replace that node with its child; that reduces the depth of all characters underneath by 1. Hence T was not optimal.

Solutiuon to 16.3-4. Claim: sum of the internal nodes' combined frequencies equals sum of the products of leaf frequencies and their depths. For example, consider this tree:



In this case, e's 9 occurrences each take 4 bits. The 9 is counted four times.
For an internal node x, the sum of the internal nodes' combined frequency of children is equal to the sum of leaf frequencies times their depth from x.

Solutiuon to 16.3-4, continued.

Proof. By structural induction.

Base case: Suppose x is an internal node both of whose children, a and b, are leaves. Then the combined frequency is

$$a.freq + b.freq = a.freq \cdot 1 + b.freq \cdot 1$$

...which is the leaf frequencies times their depths.

Inductive case 1: Suppose x is an internal node with one child being a leaf (a) and the other being itself an internal node (c); suppose that this is true for c. Let d be the combined frequency of c and d' the sum of the combined frequencies of internal nodes under c. Let $c_1, \ldots c_m$ be the leaves under c with depths (from c), $c'_1, \ldots c'_m$. Then the sum of the combined frequencies under c is

$$\begin{aligned} a.freq + d + d' &= a.freq + d + c'_1 \cdot c_1 + \ldots c'_m \cdot c_m && \text{by the ind hyp} \\ &= a.freq + c_1 + \cdot + c_m + c'_1 \cdot c_1 + \ldots + c'_m \cdot c_m \\ &= 1 \cdot a.freq + (c'_1 + 1)c_1 + \ldots (c'_m + 1)c_m \end{aligned}$$

The argument is similar in inductive case 2, where both children are themselves internal nodes. □

Generic problem: What is the best combination of items by some measure of goodness and under some constraint?

Define a **matroid** as $(S, \mathcal{I})$

- S is the items from which to choose.
- $\mathcal{I} \subseteq \mathscr{P}(S)$ is the set of combinations that satisfy the constraint
  - $\mathcal{I}$ is **hereditary**: $\forall\ B \in \mathcal{I}, \mathscr{P}(B) \subseteq \mathcal{I}$
  - $\mathcal{I}$ has the **exchange property**: $\forall\ A, B \in \mathcal{I}$, if $|A| < |B|$, then $\exists\ x \in B - A \mid A \cup \{x\} \in \mathcal{I}$.

The constraint is formally called **independence**.

Goodness is defined by a **weight function** $w : \mathcal{I} \to \mathbb{R}^+$.

The exchange property is the stand-in for the greedy choice.

Lemma extracted from proof of **Theorem 16.5** in CLRS pg 483:
If graph $G = (V, E)$ is a forest, then G contains exactly $|V| - |E|$ trees.

Proof. By induction on the size of E.
Base case. Suppose $|E| = 0$. Then each vertex is its own tree, and G has $|V| = |V| - |E|$ trees.
Inductive case. Suppose $|E| > 0$. Let $(v_1, v_2) \in E$, and let $G' = (V, E - (v_1, v_2))$. This detatches the tree containing v_1 and v_2 in G into two separate trees in G'.
By induction, G' has $|V| - |E - (v_1, v_2)|$ trees. G has one fewer tree than G', that is $|V| - |E - (v_1, v_2)| - 1 = |V| - (|E - (v_1, v_2)| + 1)| = |V| - |E|$. $\square$

Theorem 16.5 from CLRS pg 438:
If $G = (V, E)$ is an undirected graph, then the set of edges and the set of forests (acyclic subgraphs defined as subsets of edges) make a matroid.

Proof. Note $S = E$ and $\mathcal{I}$ is the set of forests of G.
(Heredity.) A subset of a forest is clearly a forest.
(Exchange property.) Suppose (V, A) and (V, B) are forests of G and $|B| > |A|$. By the lemma, (V, A) contains $|V| - |A|$ trees and (V, B) contains $|V| - |B|$ trees. There must be a tree T in (V, B) that contains an edge (u, v) where u and v are in different trees in (V, A). Thus we can add (u, v) to (V, A) without making a cycle. Hence $(V, A \cup \{(u, v)\}) \in \mathcal{I}$. $\square$

Solution 16.4-1.
$\mathcal{I} = \{X \in \mathscr{P}(S) \mid |X| \leq k\}$

(Heredity.) Suppose $B \subset S$. Further suppose $A \subset B$. By transitivity of subset, $A \subseteq S$. Also $|A| \leq |B| \leq k$, so $A \in \mathcal{I}$.

(Exchange property.) Suppose $A, B \in \mathcal{I}$ and $|A| \leq |B|$. Let $x \in B - A$. Then since $x \in B \subset A$, $x \in S$ and so $A \cup \{x\} \in S$. Moreover, $|A| < |B| \leq k$, so $|A \cup \{x\}| = |A| + 1 \leq k$, and so $A \cup \{x\} \in \mathcal{I}$.