Fractional (or Integer) Knapsack have optimal
substructure.

Only fractional problem has greedy choice property.

Statement
Let $j$ be item with maximum $\frac{V_i}{W_i}$. Then there exists an optimal solution in which you take as much of $j$ as possible.

Proof: Suppose you didn't take as much as possible of item $j$, (and the knapsack is full). Then there exists some item $k$, with $\frac{V_k}{W_k} < \frac{V_j}{W_j}$
that is in the knapsack.

Then take a wt. $\varepsilon$ piece of $k$ out, add a wt. $\varepsilon$ piece of $j$ in.

Increase knapsack value by

$$3 \frac{V_j}{W_j} - 3 \frac{V_k}{W_k} = 3 \left( \frac{V_j}{W_j} - \frac{V_k}{W_k} \right)$$

$$> 0$$

not all of $j$ is in
Alg
1. Sort items by $v_j/w_j$, renumber.
2. For $i = 1$ to $n$
   Add as much as possible of item $i$
Huffman Codes (compression)
- data on media (CD, DVD)
- data over the internet

raw data → Encoding (compression) → coded data → Decoding (code words) → raw data

Alg on media

Compression

→ lossless - encode/decode ⇒ get back original data
→ lossy - ... ⇒ get back an approx. of original data

Encode English letters

Standard encoding

ASCII - 8 bits

$2^8 = 256$

Standard encoding of $k$ symbols, you need $\lceil \lg k \rceil$ bits
encoding alg.
face
101000010100
decoding alg.
"code"

Take advantage of different frequencies of letters in English.

ETASIO...
**Different types of codes**

- **fixed length code.** Each codeword uses the same number of bits.
- **variable length code.** Codewords can use differing numbers of bits.

**Example**

<table>
<thead>
<tr>
<th>character</th>
<th>frequency</th>
<th>fixed length code</th>
<th>variable length code</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>.45</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>.13</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>c</td>
<td>.12</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>d</td>
<td>.16</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>e</td>
<td>.09</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>f</td>
<td>.05</td>
<td></td>
<td>2</td>
</tr>
</tbody>
</table>

\[
\]
### Variable Length Code

Codewords can use differing numbers of bits.

<table>
<thead>
<tr>
<th>character</th>
<th>frequency</th>
<th>fixed length code</th>
<th>variable length code</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>.45</td>
<td>000</td>
<td>0</td>
</tr>
<tr>
<td>b</td>
<td>.13</td>
<td>001</td>
<td>101</td>
</tr>
<tr>
<td>c</td>
<td>.12</td>
<td>010</td>
<td>100</td>
</tr>
<tr>
<td>d</td>
<td>.16</td>
<td>011</td>
<td>111</td>
</tr>
<tr>
<td>e</td>
<td>.09</td>
<td>100</td>
<td>1101</td>
</tr>
<tr>
<td>f</td>
<td>.05</td>
<td>101</td>
<td>1100</td>
</tr>
</tbody>
</table>

**Evaluation of code:** Expected number of bits per codeword.

**Fixed length code:** 3

**Variable length code:**

\[
.45(1) + .13(3) + .12(3) + .16(3) + .09(4) + .05(4) = 2.24
\]

**Prefix free codes:** No codeword is a prefix of any other codeword.

---

### Diagram

- **Path:** face
- **Codewords:** 110001001101
- **Decoded:** face

---

Oct 20-4:54 PM
Oct 20-5:05 PM

Oct 20-5:13 PM
Let $a$ and $b$ be two characters that are sibling leaves of maximum depth in $T$. (wlog, $f[a] \leq f[b]$ and $f[x] \leq f[y]$.)

- $f[x] \leq f[a]$ and $f[y] \leq f[b]$, since $f[x]$ and $f[y]$ are the two lowest leaf frequencies.
- Exchange the positions in $T$ of $a$ and $x$ to produce a tree $T'$.
- Exchange the positions in $T'$ of $b$ and $y$ to produce a tree $T''$.

Now look at the difference between $B(T)$ and $B(T')$

$$B(T) - B(T') = \sum_{c \in C} f(c)d_T(c) - \sum_{c \in C} f(c)d_{T'}(c)$$

$$= f[x]d_T(x) + f[a]d_T(a) - f[x]d_{T'}(x) - f[a]d_{T'}(a)$$

$$= (f[a] - f[x])(d_T(a) - d_T(x))$$

$$\geq 0,$$

Reasons for last inequality:

**Assuming**

- Each character is encoded individually
- into an integral # of bits

Huffman coding is optimal

(minimum expected # of bits transmitted)