Greedy Algorithms

• Given a set of activities \((s_i, f_i)\), we want to schedule the maximum number of non-overlapping activities.

• **GREEDY-ACTIVITY-SELECTOR** \((s, f)\)
  1. \(n = \text{length}[s]\)
  2. \(S = \{a_1\}\)
  3. \(i = 1\)
  4. **for** \(m = 2\) **to** \(n\) **do**
  5. \(\text{if } s_m \text{ is not before } f_i \text{ then}\)
  6. \(S = S \cup \{a_m\}\)
  7. \(i = m\)
  8. return \(S\)
Example

• \([1,4], [3,5], [0,6], [5,7], [3,8], [5,9], [6,10], [8,11], [8,12], [2,13], [12,14]\) -- Sorted by finish times
• \([1,4], [3,5], [0,6], [5,7], [3,8], [5,9], [6,10], [8,11], [8,12], [2,13], [12,14]\)
• \([1,4], [3,5], [0,6], [5,7], [3,8], [5,9], [6,10], [8,11], [8,12], [2,13], [12,14]\)
• \([1,4], [3,5], [0,6], [5,7], [3,8], [5,9], [6,10], [8,11], [8,12], [2,13], [12,14]\)
• \([1,4], [3,5], [0,6], [5,7], [3,8], [5,9], [6,10], [8,11], [8,12], [2,13], [12,14]\)
• \([1,4], [3,5], [0,6], [5,7], [3,8], [5,9], [6,10], [8,11], [8,12], [2,13], [12,14]\)
Why does it work?

• **THEOREM**
  Let $A$ be a set of activities and let $a_1$ be the activity with the earliest finish time. Then activity $a_1$ is in some maximum-sized subset of non-overlapping activities.

• **PROOF**
  Let $S'$ be a solution that does not contain $a_1$. Let $a'_1$ be the activity with the earliest finish time in $S'$. Then replacing $a'_1$ by $a_1$ gives a solution $S$ of the same size.

Why are we allowed to replace? Why is it of the same size?

Then apply induction! **How?**
Greedy Algorithms – Huffman Coding

- Huffman Coding Problem

**Example:** Release 29.1 of 15-Feb-2005 of **TrEMBL** Protein Database contains 1,614,107 sequence entries, comprising 505,947,503 amino acids. There are 20 possible amino acids. What is the minimum number of bits to store the compressed database?

~2.5 G bits or 300MB.

- How to improve this?
- **Information:** Frequencies are not the same.

<table>
<thead>
<tr>
<th>Amino Acid</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ala (A)</td>
<td>7.72</td>
</tr>
<tr>
<td>Arg (R)</td>
<td>5.24</td>
</tr>
<tr>
<td>Asn (N)</td>
<td>4.28</td>
</tr>
<tr>
<td>Asp (D)</td>
<td>5.28</td>
</tr>
<tr>
<td>Cys (C)</td>
<td>1.60</td>
</tr>
<tr>
<td>Gln (Q)</td>
<td>3.91</td>
</tr>
<tr>
<td>Glu (E)</td>
<td>6.54</td>
</tr>
<tr>
<td>Gly (G)</td>
<td>6.90</td>
</tr>
<tr>
<td>His (H)</td>
<td>2.26</td>
</tr>
<tr>
<td>Ile (I)</td>
<td>5.88</td>
</tr>
<tr>
<td>Leu (L)</td>
<td>9.56</td>
</tr>
<tr>
<td>Lys (K)</td>
<td>5.96</td>
</tr>
<tr>
<td>Met (M)</td>
<td>2.36</td>
</tr>
<tr>
<td>Phe (F)</td>
<td>4.06</td>
</tr>
<tr>
<td>Pro (P)</td>
<td>4.87</td>
</tr>
<tr>
<td>Ser (S)</td>
<td>6.98</td>
</tr>
<tr>
<td>Thr (T)</td>
<td>5.52</td>
</tr>
<tr>
<td>Trp (W)</td>
<td>1.18</td>
</tr>
<tr>
<td>Tyr (Y)</td>
<td>3.13</td>
</tr>
<tr>
<td>Val (V)</td>
<td>6.66</td>
</tr>
</tbody>
</table>

- **Idea:** Use shorter codes for more frequent amino acids and longer codes for less frequent ones.
Huffman Coding

2 million characters in file.


IDEA 1: Use ASCII Code
Each need at least 8 bits,
Total = 16 M bits = 2 MB

IDEA 2: Use 4-bit Codes
Each need at least 4 bits,
Total = 8 M bits = 1 MB

IDEA 3: Use Variable Length Codes

A  22  11
T  22  10
C  18  011
G  18  010
N  10  001
Y  5   00011
R  4   00010
S  4   00001
M  3   00000

How to Decode?
Need Unique decoding!
Easy for Ideas 1 & 2.
What about Idea 3?

110101101110010001100000000110
110101101110010001100000000110

Percentage Frequencies

2 million characters in file.
Length = ?
Expected length = ?
Sum up products of frequency times the code length, i.e.,
(.22x2 + .22x2 + .18x3 + .18x3 + .10x3 + .05x5 + .04x5 + .04x5 + .03x5) x 2 M bits =
3.24 M bits = .4 MB