Chapter 5, Dynamic Programming:

- Introduction and sample problems (last week Friday)
- Principles of DP (Monday)
- DP algorithms, solutions to sample problems (Today)
- Optimal BSTs (Friday)
- Finish up optimal BSTs, review for test 2 (next week Monday)
- **Test 2**, next week Wed Apr 6, *not* covering DP

Today:

- Algorithm for Knapsack
- Recursive characterization and algorithm for Longest Common Subsequence
- Recursive characterization and algorithm for Matrix Multiplication
Which of the following phrases uses the word *programming* in the same sense (or, at least, most nearly the same sense) as the phrase *dynamic programming* uses the word.

- Parallel programming
- Linear programming
- eXtreme Programming
- Pair programming
0-1 Knapsack.

Given a capacity $c$ and the value and weight of $n$ items in arrays $V$ and $W$, find a subset of the $n$ items whose total weight is less than or equal to the capacity and whose total value is maximal.

<table>
<thead>
<tr>
<th>$V$</th>
<th>20</th>
<th>15</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W$</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

$c = 7$

- set: $\{2, 3\}$, weight: 9, value: 190, exceeds capacity
- set: $\{1, 3\}$, weight: 7, value: 115, not optimal
- set: $\{0, 1, 2\}$, weight: 7, value: 125, optimal
Knapsack
Let $B[i][j]$ be the value of the best way to fill remaining knapsack capacity $i$ using only items 0 through $j$. Then $B[c][n - 1]$ is the value-solution to the entire problem, that is,

$$B[c][n - 1] = \max_{K} \sum_{j=0}^{n-1} K[j] V[j]$$

In the general case we have the choice between

$$V[j] + B[i - W[j]][j - 1]$$

value of the $j$th item

remaining capacity after taking the $j$th item

versus

$$B[i][j - 1]$$

The best way to fill the unchanged capacity with the remaining items

The best way to fill the remaining capacity with the remaining items
Knapsack

\[ B[i][j] = \begin{cases} 
0 & \text{if } j = 0 \text{ and } W[0] > i \quad (0\text{th doesn’t fit}) \\
V[0] & \text{if } j = 0 \text{ and } W[0] \leq i \quad (0\text{th fits}) \\
B[i][j - 1] & \text{if } W[j] > i \quad (j\text{th doesn’t fit}) \\
\max \left\{ V[j] + B[i - W[j]][j - 1], B[i][j - 1] \right\} & \text{otherwise} \quad (j \text{ fits})
\end{cases} \]
### Problem:

<table>
<thead>
<tr>
<th>item</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>weight</td>
<td>1</td>
<td>11</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>value</td>
<td>150</td>
<td>990</td>
<td>70</td>
<td>50</td>
<td>40</td>
</tr>
</tbody>
</table>

| 2 | 0/S | 150/S | 150/S | 150/S | 150/S | 150/S | 150/S | 220/T | 220/T | 220/T | 220/T |
| 1 | 0/S | 150/S | 150/S | 150/S | 150/S | 150/S | 150/S | 150/S | 150/S | 150/S | 150/S |
| 0 | 0/S | 150/T | 150/T | 150/T | 150/T | 150/T | 150/T | 150/T | 150/T | 150/T | 150/T |

capacities
Longest common subsequence.

*Given two sequences, find the longest subsequence that they have in common.*
Longest common subsequence

Let $L[i][j]$ be the length of the longest common subsequence of $a[:i]$ and $b[:j]$. Then $L[n][m]$ is the top-level problem.

$$L[i][j] = \begin{cases} 
0 & \text{if } i = 0 \text{ or } j = 0 \\
1 + L[i-1][j-1] & \text{if } i \neq 0 \text{ and } j \neq 0 \\
\max\{L[i][j-1], L[i-1][j]\} & \text{otherwise}
\end{cases}$$

\begin{align*}
\text{(At least one prefix is empty)} \\
\text{(Last symbols match—take it)} \\
\text{(Last symbols don’t match—skip one)}
\end{align*}
For subsequences algorithms and datastructures, the table would be:

<table>
<thead>
<tr>
<th></th>
<th>s</th>
<th>m</th>
<th>h</th>
<th>t</th>
<th>i</th>
<th>r</th>
<th>o</th>
<th>g</th>
<th>l</th>
<th>a</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>
Matrix multiplication.

Given $n + 1$ dimensions of $n$ matrices to be multiplied, find the optimal order in which to multiply the matrices, that is, find the parenthesization of the matrices that will minimize the number of scalar multiplications.

Assume the following matrices and dimensions: $A, 3 \times 5$; $B, 5 \times 10$; $C, 10 \times 2$, $D, 2 \times 3$; $E, 3 \times 4$.

$(A \times B) \times (C \times (D \times E)) \quad 3 \cdot 5 \cdot 10 + 2 \cdot 3 \cdot 4 + 10 \cdot 2 \cdot 4 + 3 \cdot 10 \cdot 4 \quad = \quad 374$

$(A \times (B \times C)) \times (D \times E) \quad 5 \cdot 10 \cdot 2 + 2 \cdot 3 \cdot 4 + 3 \cdot 5 \cdot 2 + 3 \cdot 2 \cdot 4 \quad = \quad 178$

$A \times (B \times (C \times (D \times E))) \quad 2 \cdot 3 \cdot 4 + 10 \cdot 2 \cdot 4 + 5 \cdot 10 \cdot 4 + 3 \cdot 5 \cdot 4 \quad = \quad 364$
Matrix multiplication.

\[
\begin{pmatrix}
2 & 8 \\
5 & 7
\end{pmatrix}
\begin{pmatrix}
3 & 6 \\
1 & 4
\end{pmatrix}
= 
\begin{pmatrix}
2 \cdot 3 + 8 \cdot 1 & 2 \cdot 6 + 8 \cdot 4 \\
5 \cdot 3 + 7 \cdot 1 & 5 \cdot 6 + 7 \cdot 4
\end{pmatrix}
= 
\begin{pmatrix}
14 & 44 \\
22 & 58
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 3 & 12 \\
2 & 7 & 11
\end{pmatrix}
\begin{pmatrix}
4 & 10 \\
8 & 6 \\
9 & 5
\end{pmatrix}
= 
\begin{pmatrix}
1 \cdot 4 + 3 \cdot 8 + 12 \cdot 9 & 1 \cdot 10 + 3 \cdot 6 + 12 \cdot 5 \\
2 \cdot 4 + 7 \cdot 8 + 11 \cdot 9 & 2 \cdot 10 + 7 \cdot 6 + 11 \cdot 5
\end{pmatrix}
= 
\begin{pmatrix}
136 & 88 \\
163 & 117
\end{pmatrix}
\]

\[
\begin{pmatrix}
1 & 2 & 5 \\
6 & 8 & 9
\end{pmatrix}
\begin{pmatrix}
3 \\
7 \\
4
\end{pmatrix}
= 
\begin{pmatrix}
1 \cdot 3 + 2 \cdot 7 + 5 \cdot 4 \\
6 \cdot 3 + 8 \cdot 7 + 9 \cdot 4
\end{pmatrix}
= 
\begin{pmatrix}
37 \\
110
\end{pmatrix}
\]
Matrix multiplication
Let $M[i][j]$ be the least number of scalar multiplications needed to multiply submatrices $A_i$ through $A_j$, inclusive. Then $M[0][n - 1]$ is the top-level problem.

$$M[i][j] = \begin{cases} 0 & \text{if } i = j \quad \text{(Only one matrix)} \\ \min_{i \leq k < j}\{M[i][k] + D[i]D[k + 1]D[j + 1] + M[k + 1][j]\} & \text{otherwise (Find the best way} \\
\text{to cut this series)} \end{cases}$$

Close-up of the recursive case:

$$\underbrace{M[i][k]}_{\text{minimum multiplications for}} + \underbrace{D[i]D[k + 1]D[j + 1]}_{\text{multiplications for}} + \underbrace{M[k + 1][j]}_{\text{minimum multiplications for}}$$

$$(A_i \cdots A_k)(A_{k+1} \cdots A_j)$$
Which subproblems does subproblem $M[3][7]$ depend on?

For $k$ ranging over $[3, 7)$:

$$
\begin{array}{ccccccc}
\text{k} & 3 & 4 & 5 & 6 & 5 & 6 \\
\text{subproblems} & (M[3][3], M[4][7]) & (M[3][4], M[5][7]) & (M[3][5], M[6][7]) & (M[3][6], M[7][7]) & & \\
\end{array}
$$

0 1 2 3 4 5 6 7 8
\begin{equation*}
M[i][j] = \begin{cases} 
0 & \text{if } i = j \quad \text{(Only one matrix)} \\
\min_{i \leq k < j}\{M[i][k] + D[i]D[k+1]D[j+1] + M[k+1][j]\} & \text{otherwise} \quad \text{(Find the best way to cut this series)}
\end{cases}
\end{equation*}

Problem: \( D = [10, 8, 4, 7, 3, 6, 9, 5] \)
Cell indices \((i, j)\)

Base cases

\((0,0), (1,1), (2,2), (3,3), (4,4), (5,5), (6,6)\)
\((0,1), (1,2), (2,3), (3,4), (4,5), (5,6)\)
\((0,2), (1,3), (2,4), (3,5), (4,6)\)
\((0,3), (1,4), (2,5), (3,6)\)
\((0,4), (1,5), (2,6)\)
\((0,5), (1,6)\)

Top-level problem

\((0,6)\)

Problem: \(D = [10, 8, 4, 7, 3, 6, 9, 5]\)
Coming up:

Catch up on projects…

Due Wed, Mar 30 (classtime)
Read Section 6.4
Do Exercises 6.(20, 24, 32)

Due Fri, Apr 1 (end of day)
Do Project 6.1.b as a practice problem
Take quiz (on Section 6.4)

Due Mon, Apr 4 (end of day—though a skim is recommended for Apr 1)
Read Section 6.5
(No quiz on Section 6.5)

(See Schoology for practice problems for Test 2; general study guide forthcoming…)