CS 48000: Algorithms & Data
Lecture 9
February 7, 2017
# Blueprint of DP in knapsack

<table>
<thead>
<tr>
<th>Subproblems</th>
<th>Best(i, C): max value for sack size C with items {i,i+1, ..., n}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guess</td>
<td>Whether pick item i or not</td>
</tr>
<tr>
<td>Recurrence</td>
<td>( \text{Best}(i, C) = \max(v_i + \text{Best}(i + 1, C - w_i), \text{Best}(i + 1, C)) )</td>
</tr>
<tr>
<td>Order</td>
<td>Compute \text{Best}(i,C) for i from n down to 1</td>
</tr>
<tr>
<td>Solve orig. problem</td>
<td>Return \text{Best}(1, W)</td>
</tr>
</tbody>
</table>
Chompo bar

• Given an $n \times m$ bar
• Wants to break into perfect squares with minimum number of cuts
• With each cut, can split the bar horizontally or vertically
• $2 \times 3$
  • $\rightarrow (2 \times 2, 2 \times 1)$
  • $\rightarrow (2 \times 2, 1 \times 1, 1 \times 1)$
# Blueprint of DP in chompo bar

<table>
<thead>
<tr>
<th>Subproblems</th>
<th>$C(j,k)$ : number of cuts to divide a $j \times k$ bar into perfect squares</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guess</td>
<td>First cut</td>
</tr>
<tr>
<td>Recurrence</td>
<td></td>
</tr>
<tr>
<td>Order</td>
<td></td>
</tr>
<tr>
<td>Solve orig.</td>
<td></td>
</tr>
<tr>
<td>problem</td>
<td></td>
</tr>
</tbody>
</table>
Avidan, Shamir SIGGRAPH’07
Which seam to delete?

- Define energy function ~ amount of information for each pixel
- Find minimally informative seam and delete
Pixel energy

- Energy = How much color variation around \((x_0, y_0)\)
- Energy = Magnitude of gradient
  \[ e(x_0, y_0) = e_x(x_0, y_0) + e_y(x_0, y_0) \]
  where
  \[ e_x(x_0, y_0) = |Img(x_0 + 1, y_0) - Img(x_0 - 1, y_0)| \]

\[
\begin{align*}
e_x(x_0, y_0) &= 20 \\
e_y(x_0, y_0) &= 80
\end{align*}
\]
Find least energy vertical seam

• A vertical seam goes from bottom edge to top edge

(x-1,y+1)  (x,y+1)  (x+1,y+1)

(x,y)
Identify subproblems

- $S(x,y)$: least energy to go from bottom edge to $(x,y)$
- Best seam to be deleted has to be the one with least energy among $S(1,n)$, $S(2,n)$, ..., $S(m,n)$
Optimal substructure

- Consider the optimal seam to reach \((i,j)\)
- Consider the last point \((u,v)\) on the seam before reaching \((i,j)\)
- Path from bottom to \((u,v)\) must also be optimal
Recursive relation

\[ S(x, y) = e(x, y) + \min \left\{ S(x - 1, y - 1), S(x, y - 1), S(x + 1, y - 1) \right\} \]
Evaluation order

- Initialize $S[x, 0] \leftarrow 0 \ \forall x$  // base cases
- Initialize $S[0, y] \leftarrow \infty, S[m + 1, y] \leftarrow \infty$  // sentinels
- For $y \leftarrow 1$ to $n$,  // go from bottom to top
  - For each $x$ in $\{1, \ldots, m\}$
    $$S(x, y) \leftarrow e(x, y) + \min \begin{cases} S(x - 1, y - 1) \\ S(x, y - 1) \\ S(x + 1, y - 1) \end{cases}$$
Compute final answer

• Pick best among all end points on the top row
  • $\text{best}_x \leftarrow 1$
  • For $x$ from 1 to $m$
    • If $S(x, n) < S(\text{best}_x, n)$
      • $\text{best}_x \leftarrow x$
  • Return $S(\text{best}_x, n)$
Trace back the seam

• Initialize $S[x, 0] \leftarrow 0 \ \forall x$ // base cases
• Initialize $S[0, y] \leftarrow \infty, S[m + 1, y] \leftarrow \infty$ // sentinels
• For $y \leftarrow 1 \text{ to } n$,
  • For each $x$ in $\{1, \ldots, m\}$
    • $S(x, y) \leftarrow e(x, y) + \min \begin{cases} S(x - 1, y - 1) \\ S(x, y - 1) \\ S(x + 1, y - 1) \end{cases}$
    • $\text{choice}(x, y) \leftarrow \text{either } x - 1, x, \text{ or } x + 1 \text{ depending on which one is the minimum above}$
Trace back the seam

- $x \leftarrow best_x$
- $for\ y \leftarrow n\ downto\ 1$
  - Output point $(x, y)$
  - $x \leftarrow choice[x, y]$
Text justification
Packing words into lines

- Sequence of words $w_1, \ldots, w_n$
- $w_i$ is the width of i-th word
- Want to pack words into lines in the most aesthetically pleasing way
Rule 1: no text in margin

The quick brown fox jumps over the lazy dog
Rule 2: avoid slack if possible

O, they have lived long on the alms-basket of words. I marvel thy master hath not eaten thee for a word; for thou art not so long by the head as *honorificabilitudinitatibus*: thou art easier swallowed than a flap-dragon.
Objective

• Partition words into lines so that
  • Total width of each line is at most page width $W$
    
    $$(\text{#words on line } i) - 1 + \sum_{j \text{ on line } i} w_j \leq W$$
  • Slack on line $i$
    
    $W - (\text{#words on line } i) + 1 - \sum_{j \text{ on line } i} w_j$
  • Minimize sum of slacks cubed

Spaces
Identify subproblems

• Best(n): minimum badness for typesetting first n words