Recursive Backtracking

Hitesh Boinpally
Summer 2023
Agenda

• Review
• Gambling pt. 2
• Scrabble
• P2 Preview
Agenda

• Review
• Gambling pt. 2
• Scrabble
• P2 Preview
Exhaustive Search

• Brute force algorithm!
  • Not *smart*
• Simply trying all possibilities, find the ones that work
• Often solved recursively
Exhaustive Search

• Brute force algorithm!
  • Not smart
• Simply trying all possibilities, find the ones that work
• Often solved recursively
• Two main cases
  • Found Solution: base case
  • Continue Exploring: recursive case
• Multiple recursive calls
  • One per option
• Public/Private pairs are essential
Exhaustive Search

• Brute force algorithm!
  • Not smart
• Simply trying all possibilities, find the ones that work
• Often solved recursively
• Two main cases
  • Found Solution: base case
  • Continue Exploring: recursive case
• Multiple recursive calls
  • One per option
• Public/Private pairs are essential
fourAB Visualized
Agenda

• Review
• Gambling pt. 2
• Scrabble
• P2 Preview
Dice Rolls Review

- Write a method that prints out all possible outcomes when rolling some n 6-sided dice.

Example: `diceRoll(2)`

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[1, 1]</td>
<td>[3, 1]</td>
<td>[5, 1]</td>
</tr>
<tr>
<td>[1, 2]</td>
<td>[3, 2]</td>
<td>[5, 2]</td>
</tr>
<tr>
<td>[1, 3]</td>
<td>[3, 3]</td>
<td>[5, 3]</td>
</tr>
<tr>
<td>[1, 4]</td>
<td>[3, 4]</td>
<td>[5, 4]</td>
</tr>
<tr>
<td>[1, 5]</td>
<td>[3, 5]</td>
<td>[5, 5]</td>
</tr>
<tr>
<td>[1, 6]</td>
<td>[3, 6]</td>
<td>[5, 6]</td>
</tr>
<tr>
<td>[2, 1]</td>
<td>[4, 1]</td>
<td>[6, 1]</td>
</tr>
<tr>
<td>[2, 2]</td>
<td>[4, 2]</td>
<td>[6, 2]</td>
</tr>
<tr>
<td>[2, 3]</td>
<td>[4, 3]</td>
<td>[6, 3]</td>
</tr>
<tr>
<td>[2, 4]</td>
<td>[4, 4]</td>
<td>[6, 4]</td>
</tr>
<tr>
<td>[2, 5]</td>
<td>[4, 5]</td>
<td>[6, 5]</td>
</tr>
<tr>
<td>[2, 6]</td>
<td>[4, 6]</td>
<td>[6, 6]</td>
</tr>
</tbody>
</table>
Dice Sums

• Write a method that prints out all outcomes that sum to a particular value when rolling some n 6-sided dice.

Example: diceRoll(2, 7)

[1, 6]
[2, 5]
[3, 4]
[4, 3]
[5, 2]
[6, 1]
Should we consider all paths?

<table>
<thead>
<tr>
<th>chosen</th>
<th>available</th>
<th>desired sum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3 dice</td>
<td>5</td>
</tr>
</tbody>
</table>

1, 1, 1
1, 1, 2
1, 1, 3
1, 1, 4
1, 1, 5
1, 1, 6
1, 6, 1
1, 6, 2
...
Should we consider all paths?

<table>
<thead>
<tr>
<th>chosen</th>
<th>available</th>
<th>desired sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>3 dice</td>
<td>5</td>
</tr>
</tbody>
</table>

1 dice: 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 3, 1, 1, 4, 1, 1, 5, 1, 1, 6

2 dice: 1, 1, 2, 1, 1, 3, 1, 1, 4, 1, 1, 5, 1, 1, 6, 1, 6, 1, 1, 6, 2, ...
Should we consider all paths?

<table>
<thead>
<tr>
<th>chosen</th>
<th>available</th>
<th>desired sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>3 dice</td>
<td>5</td>
</tr>
</tbody>
</table>

1, 2 dice 2 dice 3 dice 4 dice 5 dice 6 dice

1, 1 die 1, 2 die 1, 3 die 1, 4 die 1, 5 die 1, 6 die

1, 1, 1 1, 1, 2 1, 1, 3 1, 1, 4 1, 1, 5 1, 1, 6

1, 6, 1 1, 6, 2...
Should we consider all paths?

<table>
<thead>
<tr>
<th>chosen</th>
<th>available</th>
<th>desired sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>3 dice</td>
<td>5</td>
</tr>
</tbody>
</table>

1, 1, 1  1, 1, 2  1, 1, 3

1, 6, 1  1, 6, 2  ...

(Visualization courtesy of Hunter Schafer)
Exhaustive Search

• Brute force algorithm!
  • Not **smart**
• Simply trying all possibilities, find the ones that work
• Often solved recursively

• Two main cases
  • **Found Solution:** base case
  • **Continue Exploring:** recursive case

• Multiple recursive calls
  • One per option
• Public/Private pairs are essential
Recursive Backtracking

• Brute force algorithm!
  • Not smart
• Simply trying all possibilities, find the ones that work
• Often solved recursively

• Two Three main cases
  • Found Solution: base case
  • Continue Exploring: recursive case
  • Dead End: Implicit case to stop exploring

• Multiple recursive calls
  • One per option
• Public/Private pairs are essential
Agenda

• Review
• Gambling pt. 2
• Scrabble
• P2 Preview
Agenda

• Review
• Gambling pt. 2
• Scrabble
• P2 Preview
P2 Preview

• Utilize concepts learned today
  • Parts of diceSum and Scrabble will be useful
• Lots of helper classes
  • Make sure you understand how to use each of them!
P2 Preview

• Utilize concepts learned today
  • Parts of diceSum and Scrabble will be useful
• Lots of helper classes
  • Make sure you understand how to use each of them!
• Initial submission numbers have been low
• Remember that to be making good progress you should be submitting a (at least) behaviorally fully functional assignment initially
  • Then utilize the resubmission to improve your score