stdint.h

#define int64_t unsigned 64-bit int
#define int32_t signed 32-bit int

unsigned long

range

0h 2^64 - 1

-2^31 h 2^31 - 1

int
1, 23, 37 stamps (unlimited supply of each)

Make 50¢ using fewest possible stamps

\[ 50 = \underbrace{23 + 23 + 1 + 1 + 1}_6 \text{ stamps} \]

In general: if \( v_1, v_2, \ldots, v_k \) is shortest list with \( v_i \in \{1, 23, 37\} \) and \( \sum v_i = n \)

if it weren't - if

then \( v_1, \ldots, v_{k-1} \) is shortest list with sum \( n - v_k \)

\( u_1, \ldots, u_k \) is better \((\leq k-1)\)

- way to make \( n - v_k \) cents

then \( u_1, \ldots, u_k, v_k \)

is better than best way to make \( n \) cents

\[ \text{num}(n) = 1 + \min_{n \geq 0} \left( \text{num}(n-37), \text{num}(n-23), \text{num}(n-1) \right) \]

\[ \text{num}(0) = 0 \]
Counting Chomp States

How many possible states in 3x4 Chomp?

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Does this depend on # of states in 3x3 Chomp?

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For h x w Chomp, # states:

- # h x (w-1) states that end in 0
- # h x (w-1) states that end in 1
- # h x (w-1) states that end in 2
- # h x (w-1) states that end in h

So need to know # h x i states that end in k

\[
\text{count}(i,k) = \text{# h x i Chomp states w/ } k \text{ in rightmost column}
\]

\[
= \text{# h x (i-1) states w/ } k \text{ in rightmost } + \text{# h x (i-1) states w/ } k+1 \text{ in rightmost } + \text{# h x (i-1) states w/ } h \text{ in rightmost}
\]

\[
= \sum_{j=k}^{h} \text{count}(i-1,j)
\]

**count(1,k)**

\[
\text{count}(1,k) = 1
\]

\[
\text{count}(w+1,h+1)
\]

\[
\text{init: } \text{count}(1-1,0) = 1 \text{ for } c = 0 \ldots h
\]

\[
\text{for } c = 2 \text{ to } w
\]

\[
\text{for } i = 0 \text{ to } h
\]

\[
\text{count}(c-1,i) = \sum_{j=0}^{h} \text{count}(c-1-1,j)
\]

(Recursively, bottom-up, right-to-left)

\[
O(w h^2)
\]
Dynamic Programming Page 4

```
return sum of last row
```

(or extend table one more row
and return 1st entry in that row, which is...
Image Resizing by Seam Carving

```
energy

0 1 2 3 4 5
0 1 1 1 4 1 2
1 3 6 5 1 0 3
2 4 3 2 2 0 3
3 2 1 6 5 1 3
4 1 0 2 1 0 0
```

Want seam with
Let $v_0, v_1, ..., v_k$ be min-cost path $s \rightarrow u$

Then $s, ..., v_{k-1}$ is

let $d[u, k] =$ cost of min-cost path $s \rightarrow u$ using $\leq k$ edges

Then