Dynamic Programming
Matrix
\[ A_1 \cdot A_2 \cdot A_3 \]
\[(A_1 \cdot A_2) \cdot A_3 \quad A_1 \cdot (A_2 \cdot A_3)\]
\((A_1 \cdot A_2) \cdot A_3\)
\[(A_1 \cdot A_2) \cdot A_3\]

\[10 \cdot 100 \cdot 5 + 10 \cdot 5 \cdot 50\] operations
\[ A_1 \cdot A_2 \cdot A_3 \]

\[ 10 \cdot 100 \cdot 50 = 50,000 \]
\[ A_1 \cdot A_2 \cdot A_3 \]

\[ 100 \cdot 5 \cdot 50 + 10 \cdot 100 \cdot 50 \]

operations
Order matters
(for efficiency)
How many ways to compute?

$A_1 A_2 A_3 \ldots A_n$

$P(n) = \# \text{ of different ways to mult the } n \text{ matrices.}$
How many ways to compute?

\[ P(n) = P(1)P(n-1) + P(2)P(n-2) + \ldots + P(n-1)P(1) \]

\[ \approx 4^n \]
How many ways to compute?

\[ A_1 A_2 A_3 \ldots A_n \]

\[ A_1 A_2 A_3 \ldots A_n \]
How many ways to compute?

$A_1 A_2 A_3 \ldots A_n$

$A_1 A_2 A_3 \ldots A_n$

$A_1 A_2 A_3 \ldots A_n$
How do we solve it?

- identify smaller instances of the problem
- devise method to combine solutions
- small # of different subproblems
  - solved them in the right order
Optimal way to compute $A_1 A_2 A_3 A_4 \ldots A_n$.

$B(1,n) = \text{best way to multiply } A_1 \ldots A_n$.

$B(1,n) = B(1,k) + B(k+1,n) + \left\lceil \frac{n - c_k \cdot c_n}{c_k} \right\rceil$

how many choices are there for $k^*$?
optimal way to compute

\[ A_1 A_2 A_3 A_4 \ldots A_n \]
optimal way to compute

\[ A_1 A_2 A_3 A_4 \ldots A_n \]

\[ B[1,n] \]

\[ B[1,1] \]
\[ B[2,n] \]

\[ R_1 C_1 C_n \]
optimal way to compute

\[
\begin{align*}
A_1 A_2 A_3 A_4 \ldots A_n
\end{align*}
\]
\[ B(i, i) = 1 \]

\[ B(1, n) = \min \{ \ldots \} \]
\[
B(i, i) = 0
\]

\[
B(1, n) = \min \left\{ \begin{array}{l}
B(1, 1) + B(2, n) + r_1 c_1 c_n \\
B(1, 2) + B(3, n) + r_1 c_2 c_n \\
\vdots \\
B(1, n - 1) + B(n, n) + r_1 c_{n-1} c_n
\end{array} \right. 
\]
\( B(i, j) = \)

\[
\begin{cases}
0 & \text{if } i = j \\
\min_k \{ B(i, k) + B(k + 1, j) + r_i c_k c_j \}
\end{cases}
\]
which order to solve?
\[
B(i, j) = \begin{cases} 
0 & \text{if } i = j \\
\min_k \{ B(i, k) + B(k + 1, j) + r_i c_k c_j \} & \text{otherwise}
\end{cases}
\]
\[ B(1, 2) = \min \left\{ \frac{B(1, 1) + B(2, 2) + r_1 \cdot c_1 \cdot c_2}{30.35.15} \right\} \]
\[
B(i, j) = \begin{cases} 
0 & \text{if } i = j \\
\min_k \{B(i, k) + B(k + 1, j) + r_i c_k c_j\} & \text{otherwise}
\end{cases}
\]
\[ B(i, j) = \begin{cases} 
0 & \text{if } i = j \\
\min_k \{B(i, k) + B(k + 1, j) + r_i c_k c_j\} & \text{otherwise}
\end{cases} \]
\[ B(i, j) = \begin{cases} 
0 & \text{if } i = j \\
\min_k \{ B(i, k) + B(k + 1, j) + r_i c_k c_j \} & \text{otherwise}
\end{cases} \]
\[ C(1, 6) = \min \begin{cases} 
  k = 1 & C'(1, 1) + C'(2, 6) + r_1 c_1 c_6 \\
  k = 2 & C'(1, 2) + C'(3, 6) + r_1 c_2 c_6 \\
  k = 3 & C'(1, 3) + C'(4, 6) + r_1 c_3 c_6 \\
  k = 4 & C'(1, 4) + C'(5, 6) + r_1 c_4 c_6 \\
  k = 5 & C'(1, 5) + C'(6, 6) + r_1 c_5 c_6 
\end{cases} \]
\[
C(1, 6) = \min \begin{cases} 
  k = 1 & 0 + 10500 + 30 \cdot 35 \cdot 25 \\
  k = 2 & 15750 + 5375 + 30 \cdot 15 \cdot 25 \\
  k = 3 & 7875 + 3500 + 30 \cdot 5 \cdot 25 \\
  k = 4 & 9375 + 5000 + 30 \cdot 10 \cdot 25 \\
  k = 5 & 11875 + 0 + 30 \cdot 20 \cdot 25 
\end{cases}
\]
\[ C(1, 6) = \min \begin{cases} \quad k = 1 & 0 + 10500 + 26250 \\ \quad k = 2 & 15750 + 5375 + 11250 \\ \quad k = 3 & 7875 + 3500 + 3750 \\ \quad k = 4 & 9375 + 5000 + 7500 \\ \quad k = 5 & 11875 + 0 + 15000 \end{cases} \]
<table>
<thead>
<tr>
<th></th>
<th>30</th>
<th>35</th>
<th>35</th>
<th>15</th>
<th>15</th>
<th>5</th>
<th>5</th>
<th>10</th>
<th>10</th>
<th>20</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>15125</td>
<td>10500</td>
<td>5375</td>
<td>3500</td>
<td>10<em>20</em>25 = 5000</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>11875</td>
<td>7125</td>
<td>2500</td>
<td>0</td>
<td>5<em>10</em>20 = 1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>9375</td>
<td>4375</td>
<td>0</td>
<td>0</td>
<td>15<em>5</em>10 = 750</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>7875</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>35<em>15</em>5 = 2625</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30<em>35</em>15 = 15750</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Matrix-chain-mult(p)

initialize array m[x,y] to zero
Matrix-chain-mult(p)

initialize array m[x,y] to zero
starting at diagonal, working towards upper-left
compute m[i,j] according to
\[
\begin{cases}
0 & \text{if } i = j \\
\min_k \left\{ B(i, k) + B(k+1, j) + r_i c_k c_j \right\}
\end{cases}
\]

\[\Theta(n^3) \rightarrow O(n^2) \rightarrow \Theta(n \log n)\]
running time?

initialize array \( m[x,y] \) to zero

starting at diagonal, working towards upper-left

compute \( m[i,j] \) according to

\[
\begin{cases}
0 \text{ if } i = j \\
\min_k \{ B(i, k) + B(k + 1, j) + r_i c_k c_j \}
\end{cases}
\]
It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to heaven, we were all going direct the other way - in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.
It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to heaven, we were all going direct the other way - in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.
First rule of typesetting

never print in the margin!

are simply not allowed
It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to heaven, we were all going direct the other way — in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.
It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to heaven, we were all going direct to the other way - in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.
It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness, it was the epoch of belief, it was the epoch of incredulity, it was the season of Light, it was the season of Darkness, it was the spring of hope, it was the winter of despair, we had everything before us, we had nothing before us, we were all going direct to heaven, we were all going direct the other way - in short, the period was so far like the present period, that some of its noisiest authorities insisted on its being received, for good or for evil, in the superlative degree of comparison only.
Typesetting problem

input: \( \binom{M}{w_1, w_2, w_3 \ldots, w_n} \) \( \text{lenghs of the wires} \)

output: \( (w_1, \ldots, w_{\ell_1}), (w_{\ell_1+1}, \ldots, w_{\ell_2}), \ldots, \ldots, (w_{\ell_n}, \ldots, w_n) \)

such that
Typesetting problem

input: \[ W = \{ w_1, w_2, w_3, \ldots, w_n \} \quad M \]

output: \[ L = (w_1, \ldots, w_{\ell_1}), (w_{\ell_1+1}, \ldots, w_{\ell_2}), \ldots, (w_{\ell_x+1}, \ldots, w_n) \]

such that
Typesetting problem

input: \( W = \{w_1, w_2, w_3, \ldots, w_n\} \)

output:

\[ M \]

such that

\[ L = \left( w_1, \ldots, w_{\ell_1} \right), \left( w_{\ell_1+1}, \ldots, w_{\ell_2} \right), \ldots, \left( w_{\ell_{x+1}}, \ldots, w_n \right) \]

\[ c_i = \left( \sum_{j=\ell_i+1}^{\ell_{i+1}} |w_j| \right) + (\ell_{i+1} - \ell_i - 1) \]

such that

\[ c_i \leq M \quad \forall i \]

\[ \min \sum (M - c_i)^2 \]
define the right variable:

Best $n$: smallest $(penalty)^2$ for typesetting $n$ words
Imagine optimal solution
Imagine optimal solution

\[ \text{Best}_n = \text{Best}_{\text{full}-1} + (S_{\text{full}, n})^2 \]
Some word has to be the first-word-of-last-line (fwoll)
Imagine optimal solution

\[ w_\ell \approx S_{\ell,n} \quad \text{last line} \]

\text{fwoll is } w_\ell \quad \text{slack when line starts with } w_\ell
Imagine optimal solution

\[
\text{bwoll is } \quad w_{\ell} \quad \text{slack when line starts with } \quad w_{\ell}
\]

\[
\text{last line}
\]

\[
\text{BEST}_n = \text{BEST}_{\ell-1} + S_{\ell,n}^2
\]
How many candidates are there for the full?
Is $w_1$ well?

there is no slack (no solution even) because words go beyond edge!

define $S_{1,n} = \infty$ if this happens
Is $w_2$ floppy?

$w_1$

$w_2$

$S_{2,n} = \infty$

$\text{Best}_1 + (S_{2,n})^2$
$V_1, W_2, W_5$

$B_{est_2} + (s_{3,17})^2$
Is $w_j$ well?
Which word is *fwoll*?

\[ \text{BEST}_n = \min \{ \text{Best}, \, t, \, s \} \]
Which word is fwoll?

\[
\text{BEST}_n = \min \left\{ \begin{array}{c}
\text{BEST}_0 + S_{1,n}^2 \\
\text{BEST}_1 + S_{2,n}^2 \\
\text{BEST}_2 + S_{3,n}^2 \\
\vdots \\
\text{BEST}_{\ell-1} + S_{\ell,n}^2 \\
\vdots \\
\text{BEST}_{n-1} + S_{n,n}^2 
\end{array} \right\}
\]
How to compute \( S_{i,j} \) slack when line starts with \( w_i \) and ends \( w_j \)
Simplest case

slack when line starts with $w_i$ and ends $w_i$

$S_{1,1} = M - w_i$
Simplest case

slack when line starts with $w_i$ and ends $w_2$

$S_{1,2} = S_{1,1} - 1 - w_2$
how to compute $S_{i,j}$

slack when line starts with $w_i$ and ends $w_j$
```java
int infty = M*M*2;

// compute S_ij
int S[][] = new int[n+1][n+1];
for(int i=1;i<=n;i++) {
    S[i][i] = M - lens[i];
    for(int j=i+1; j<=n; j++) {
        S[i][j] = S[i][j-1] - lens[j] - 1;
        if (S[i][j]<0) {
            while(j<=n) { S[i][j++] = infty; }
        }
    }
}
```
Typesetting algorithm

make table for $S_{i,j}$
Typesetting algorithm

make table for $S_{i,j}$

for $i=1$ to $n$

$$\text{best}[i] = \min\{ \text{best}[j] + S[j+1][i]^2 \}$$

```java
// compute best_0,...,best_n
int best[] = new int[n+1];
int choice[] = new int[n+1];
best[0] = 0;
for (int i=1; i<=n; i++) {
    int min = infty;
    int ch = 0;
    for (int j=0; j<i; j++) {
        int t = best[j] + S[j+1][i]*S[j+1][i];
        if (t<min) { min = t; ch = j; }
    }
    best[i] = min;
    choice[i] = ch;
}
```
Example

It was the best of times, it was the worst of times; it was the age of wisdom, it was the age of foolishness; it was the epoch of belief, it was the epoch of incredulity; it was the season of
first step: make $S_{i,j}$

\[
S_{i,j} = S_{i,j-1} - 1 - |w_j|
\]

\[
S_{i,i} = M - |w_i|
\]

\[
S_{i,1} = 42 - 2
\]
First step: make $S_{i,j}$

<p>| | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>36</td>
<td>32</td>
<td>27</td>
<td>24</td>
<td>17</td>
<td>14</td>
<td>10</td>
<td>6</td>
<td>0</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td>39</td>
<td>35</td>
<td>30</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>6</td>
<td>2</td>
</tr>
</tbody>
</table>

$$S_{i,i} = M - |w_i|$$

$$S_{i,j} = S_{i,j-1} - 1 - |w_j|$$

$M = 42$

$S_{2,2} = 42 - 3 = 39$

$S_{2,3} = S_{2,2} - 1 - 3 = 35$

$S_{2,4} = S_{2,3} - 1 - 4 = 30$
was the best
First step: make \( S_{i,j} \)

\[
\begin{array}{cccccccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\
1 & 40 & 36 & 32 & 27 & 24 & 17 & 14 & 10 & 6 & 0 & 99 & 99 & 99 \\
2 & 39 & 35 & 30 & 27 & 20 & 17 & 13 & 9 & 3 & 0 & 99 & 99 & \\
3 & & & & & & & & & & & & & \\
\end{array}
\]

\[
S_{i,i} = M - |w_i|
\]

\[
S_{i,j} = S_{i,j-1} - 1 - |w_j|
\]
second step: compute

\[ \text{Best}_i = \text{Best}_0 + (S_{1,1})^2 = 0 + 40^2 = 1600 \]

\[ \text{BEST}_i = \min_{j=0}^{i-1} \left\{ \text{BEST}_j + S^2_{j+1,i} \right\} \]
\[
\text{BEST}_i = \min_{j=0}^{i-1} \{ \text{BEST}_j + S_{j+1,i}^2 \}
\]
\[ \text{BEST}_i = \min_{j=0}^{i-1} \{ \text{BEST}_j + S_{j+1,i}^2 \} \]
\[
BEST_i = \min_{j=0}^{i-1} \{BEST_j + S_{j+1,i}^2\}
\]

It was the
\[
\text{BEST}_i = \min_{j=0}^{i-1} \{ \text{BEST}_j + S_{j+1,i}^2 \}
\]

```
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>best</td>
<td>0</td>
<td>1600</td>
<td>1296</td>
<td>1024</td>
<td>729</td>
<td>576</td>
<td>289</td>
<td>196</td>
<td>100</td>
<td>36</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>choice</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>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
```

It was the best

```
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>36</td>
<td>32</td>
<td>27</td>
<td>24</td>
<td>17</td>
<td>14</td>
<td>10</td>
<td>6</td>
<td>0</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td>2</td>
<td>39</td>
<td>35</td>
<td>30</td>
<td>27</td>
<td>20</td>
<td>17</td>
<td>13</td>
<td>9</td>
<td>3</td>
<td>0</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
</tbody>
</table>
```
\[
\text{BEST}_i = \min_{j=0}^{i-1} \{ \text{BEST}_j + S_{j+1,i}^2 \}
\]
\[
\text{BEST}_i = \min_{j=0}^{i-1} \{ \text{BEST}_j + S_{j+1,i}^2 \}
\]

\[
\text{best} = [0, 1600, 1296, 1024, 729, 576, 289, 196, 100, 36, 0, 0, 0, 0]
\]

\[
\text{choice} = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
\]

It was the best of times, it was the worst of times. It was the best of times, it was the worst of times.
\[
\text{BEST}_i = \min_{j=0}^{i-1} \{ \text{BEST}_j + S_{j+1,i}^2 \}
\]

```
<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1600</th>
<th>1296</th>
<th>1024</th>
<th>729</th>
<th>576</th>
<th>289</th>
<th>196</th>
<th>100</th>
<th>36</th>
<th>0</th>
</tr>
</thead>
</table>
```

It was the best of times, it was the worst of

```
<table>
<thead>
<tr>
<th>choice</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
</table>
```

Best_{11} = \min \{ ... \}
It was the best of times, it was the worst of times.

$$\text{BEST}_i = \min_{j=0}^{i-1} \left\{ \text{BEST}_j + S_{j+1,i}^2 \right\}$$

<table>
<thead>
<tr>
<th>best</th>
<th>0</th>
<th>1600</th>
<th>1296</th>
<th>1024</th>
<th>729</th>
<th>576</th>
<th>289</th>
<th>196</th>
<th>100</th>
<th>36</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>choice</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>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

$$\text{BEST}_{11} = \min \left\{ \text{BEST}_{10} + S_{11,11}^2, \text{BEST}_9 + S_{10,11}^2, \text{BEST}_8 + S_{9,11}^2, \text{BEST}_7 + S_{8,11}^2, \text{BEST}_6 + S_{7,11}^2, \ldots \right\}$$
It was the best of times, it was the worst of times.

\[
\text{best}_i = \min_{j=0}^{i-1} \{ \text{best}_j + S_{j+1,i}^2 \}
\]

<table>
<thead>
<tr>
<th>choice</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>best</td>
<td>0</td>
<td>1600</td>
<td>1296</td>
<td>1024</td>
<td>729</td>
<td>576</td>
<td>289</td>
<td>196</td>
<td>100</td>
<td>36</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\text{best}_{11} = \min \left\{ \begin{array}{l}
\text{best}_{10} + S_{11,11}^2 \\
\text{best}_9 + S_{10,11}^2 \\
\text{best}_8 + S_{9,11}^2 \\
\text{best}_7 + S_{8,11}^2 \\
\text{best}_6 + S_{7,11}^2 \\
\end{array} \right\}
\]
It was the best of times,

it was the worst of 

**BEST**$_i$ = \( \min_{j=0}^{i-1} \{ BEST_j + S_{j+1,i}^2 \} \)

<table>
<thead>
<tr>
<th>best</th>
<th>0</th>
<th>1600</th>
<th>1296</th>
<th>1024</th>
<th>729</th>
<th>576</th>
<th>289</th>
<th>196</th>
<th>100</th>
<th>36</th>
<th>0</th>
<th>818</th>
</tr>
</thead>
<tbody>
<tr>
<td>choice</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>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**BEST**$_{11}$ = \( \min \) \( \{ \)

- BEST$_{10}$ + $S_{11,11}^2$
- BEST$_9$ + $S_{10,11}^2$
- BEST$_8$ + $S_{9,11}^2$
- BEST$_7$ + $S_{8,11}^2$
- BEST$_6$ + $S_{7,11}^2$

...
$BEST_i = \min_{j=0}^{i-1} \{BEST_j + S^2_{j+1,i}\}$

<table>
<thead>
<tr>
<th>choice</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>6</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>best</td>
<td>0</td>
<td>1600</td>
<td>1296</td>
<td>1024</td>
<td>729</td>
<td>576</td>
<td>289</td>
<td>196</td>
<td>100</td>
<td>36</td>
<td>0</td>
<td>818</td>
<td>545</td>
</tr>
</tbody>
</table>

It was the best of times, it was the worst of times, it was...
It was the best of times, it
was the worst of times, it

\[
\text{BEST}_i = \min_{j=0}^{i-1} \{ \text{BEST}_j + S_{j+1,i}^2 \}
\]

\[
\begin{array}{cccccccccccccc}
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 \\
\hline
\text{best} & 0 & 1600 & 1296 & 1024 & 729 & 576 & 289 & 196 & 100 & 36 & 0 & 818 & 545 \\
\text{choice} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6 & 6 & 0
\end{array}
\]

\[
\text{BEST}_{13} = \min \left\{ \begin{array}{l}
\text{BEST}_{12} + S_{13,13}^2 \\
\text{BEST}_{11} + S_{12,13}^2 \\
... \\
\text{BEST}_{7} + S_{8,13}^2 \\
\text{BEST}_{6} + S_{7,13}^2 \\
\end{array} \right\}
\]
```plaintext
0 best: 0 ch 0
1 best: 1600 ch 0
2 best: 1296 ch 0
3 best: 1024 ch 0
4 best: 729 ch 0
5 best: 576 ch 0
6 best: 289 ch 0
7 best: 196 ch 0
8 best: 100 ch 0
9 best: 36 ch 0
10 best: 0 ch 0
11 best: 818 ch 6
12 best: 545 ch 6
13 best: 452 ch 7
14 best: 340 ch 7
15 best: 244 ch 8
16 best: 164 ch 8
17 best: 117 ch 9
18 best: 37 ch 9
19 best: 16 ch 10
20 best: 0 ch 10
21 best: 509 ch 14
22 best: 413 ch 15
23 best: 344 ch 15
24 best: 133 ch 17
25 best: 118 ch 17
26 best: 62 ch 18
27 best: 32 ch 19
28 best: 4 ch 20
29 best: 444 ch 23
30 best: 348 ch 23
31 best: 277 ch 24
32 best: 197 ch 24
33 best: 149 ch 24
34 best: 87 ch 26
35 best: 66 ch 26
36 best: 446 ch 31
37 best: 377 ch 31
38 best: 297 ch 32
39 best: 233 ch 32
```
It was the best of times, it was the worst of times, it was the age of wisdom, it was the age of foolishness.
try {
    BufferedReader bin = new BufferedReader(new FileReader(args[0]));
    String line = bin.readLine();
    String[] words = line.split(" ");
    int n = words.length;
    int M = Integer.parseInt(args[1]);
    int[][] S = new int[n+1][n+1];
    for(int i=1; i<=n; i++) {
        S[i][i] = M - lens[i];
        for(int j=i+1; j<=n; j++) {
            S[i][j] = S[i][j-1] - lens[j] - 1;
            if (S[i][j]<0) {
                while(j<=n) { S[i][j++] = infty; }
            }
        }
    }
}
for(int i=1;i<=n; i++) {
    lens[i] = words[i-1].length();
    if (lens[i]>M) {
        System.out.println("word too long");
        System.exit(1);
    }
}

int infty = M*M*2;

// compute S_ij
int S[][] = new int[n+1][n+1];
for(int i=1;i<=n;i++) {
    S[i][i] = M - lens[i];
    for(int j=i+1; j<=n; j++) {
        S[i][j] = S[i][j-1] - lens[j] - 1;
        if (S[i][j]<0) {
            while(j<=n) { S[i][j++] = infty; }
        }
    }
}

// compute best_0,...,best_n
int best[] = new int[n+1];
int choice[] = new int[n+1];
best[0] = 0;
for(int i=1;i<=n;i++) {
    int min = infty;
    int ch  = 0;
    for(int j=0;j<i;j++) {
        int t = best[j] + S[j+1][i]*S[j+1][i];
        if (t<min) { min = t; ch = j; }
    }
    best[i] = min;
    choice[i] = ch;
}
```java
for(int i=1;i<=n; i++) {
    lens[i] = words[i-1].length();
    if (lens[i]>M) {
        System.out.println("word too long");
        System.exit(1);
    }
}

int infty = M*M*2;

// compute S_ij
int S[][] = new int[n+1][n+1];
for(int i=1;i<=n;i++) {
    S[i][i] = M - lens[i];
    for(int j=i+1; j<=n; j++) {
        S[i][j] = S[i][j-1] - lens[j] - 1;
        if (S[i][j]<0) {
            while(j<=n) { S[i][j++] = infty; }
        }
    }
}
```
for(int i=1;i<=n; i++) {
    lens[i] = words[i-1].length();
    if (lens[i]>M) {
        System.out.println("word too long");
        System.exit(1);
    }
}

int infty = M*M*2;
// compute S_ij
int S[][] = new int[n+1][n+1];
for(int i=1;i<=n;i++) {
    S[i][i] = M - lens[i];
    for(int j=i+1; j<=n; j++) {
        S[i][j] = S[i][j-1] - lens[j] - 1;
        if (S[i][j]<0) {
            while(j<=n) { S[i][j++] = infty; }
        }
    }
}

// compute best_0,...,best_n
int best[] = new int[n+1];
int choice[] = new int[n+1];
bef[0] = 0;
for(int i=1;i<=n;i++) {
    int min = infty;
    int ch  = 0;
    for(int j=0;j<i;j++) {
        int t = best[j] + S[j+1][i]*S[j+1][i];
        if (t<min) { min = t; ch = j; }
    }
    best[i] = min;
    choice[i] = ch;
}
// backtrack to output linebreaks
int end = n;
int start = choice[end]+1;
String lines[] = new String[n];
int cnt = 0;
while (end>0) {
    StringBuffer buf = new StringBuffer();
    for(int j=start; j<=end; j++) {
        buf.append(words[j-1] + " ");
    }
    lines[cnt++] = buf.toString();
    end = start-1;
    start = choice[end]+1;
}
Gerrymander
gerrymander problem

given:

output:
gerrymander problem

given: \( m A_1, A_2, \ldots, A_n \)

\[ n \text{ is even} \]

output: \( D_1, D_2 \)

such that \( |D_1| = |D_2| \)

\[
A(D_1) > \frac{mn}{4}
\]

\[
A(D_2) > \frac{mn}{4}
\]

or “failure” if no such solution is possible
Example
imagine very last precinct and how it is assigned:

\[ \frac{D_1}{j, x, y} \]

\[ P_n \]

\[ D_2 \]

\[ n-j, x, y \]

\[ \text{assign to } D_1 \]

\[ \text{assign to } D_2 \]

\[ A \text{ people vote for } A \text{ in } D_1 \]

\[ A \text{ people vote for } A \text{ in } D_2 \]
Gerrymander

\[ S_{j,k,x,y} = \]
Gerrymander

\[ S_{j,k,x,y} = \text{there is a split of first } j \text{ precincts in which } |D_1|=k \text{ and } \]
\[ x \text{ people in } D_1 \text{ vote } A \]
\[ y \text{ people in } D_2 \text{ vote } A \]

\[ S_{j-1,k,x,y-a} \quad \text{or} \quad S_{j-1,k-1,x-a,j,y} \]

true if placing point into \( D_2 \)

place \( p \) into \( D_1 \)
Gerrymander(P,A,m)

initialize array S[0,0,0,0]

$S_{j,k,x,y} = S_{j-1,k-1,x-A_j,y} \lor S_{j-1,k,x,y-A_j}$
Gerrymander\((P, A, m)\)

initialize array \(S[0, o, o, o]\)

for \(j = 1, \ldots, n\)

for \(k = 1, \ldots, j\)

for \(x = 0, \ldots, jm\)

for \(y = 0, \ldots, jm\)

fill table according to equation

search for true entry at \(S[n, n/2, >mn/4, >mn/4]\)

\[ S_{j,k,x,y} = S_{j-1,k-1,x-A_j,y} \lor S_{j-1,k,x,y-A_j} \]