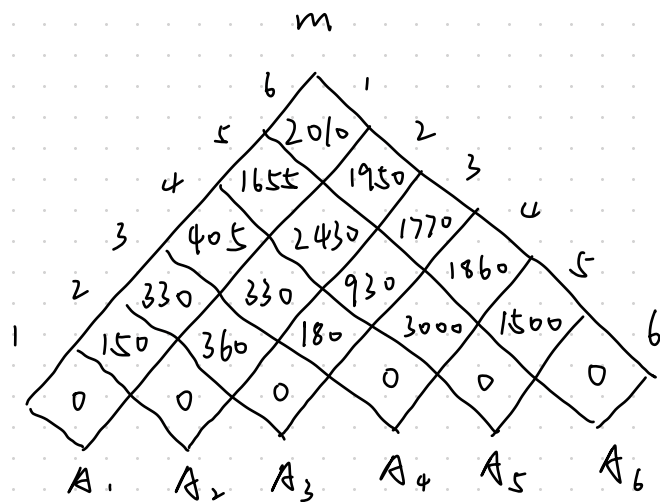


15.2-1

采用自底向上的动态规划



此时，方案为

$$((5 \times 10) (10 \times 3)) ((3 \times 12) (12 \times 5)) ((5 \times 50) (50 \times 6))$$

15-2

算法伪码如下. 其中字符串 s 的下标由 1 开始.

$F(s)$:

1. $\text{max_len} = 0$
2. $\text{len} = s.\text{length}$
3. let $m[\text{len}][\text{len}]$ be a new table
4. for $i = 1$ to len :
5. $m[i][i] = \text{True}$
6. for $l = 2$ to len :
7. for $i = 1$ to $\text{len} - l + 1$:
8. $j = i + l - 1$
9. $m[i][j] = (m[i+1][j-1] \text{ and } (s[i] == s[j]))$
10. if $m[i][j]$ and $j - i + 1 > \text{max_len}$:
11. $\text{max_len} = j - i + 1$
12. return max_len

算法思路如下:

① 若 $s[i:j]$ 是回文串, 则 $s[i+1:j-1]$ 也是回文串

② 若 $s[i+1:j-1]$ 不是回文串, 则 $s[i:j]$ 也不是回文串

③ 则有状态转移方程

$$m[i][j] = \begin{cases} \text{True} & i = j \\ s[i] == s[j] & i+1 = j \\ s[i] == s[j] \text{ and } m[i+1][j-1] & i+1 < j \end{cases}$$

④ 有以上状态方程后, 就可以写出自底向上的伪代码.

复杂度:

空间复杂度为 $O(n^2)$

时间复杂度为 $O(n^3)$

15-4

算法伪码如下,

define $MyCal(i, j) = M - j + i - \sum_{k=i}^j l[k]$

$F(l, n, M)$:

1. let $p[0 \dots n]$ be a new array
2. let $m[n][n]$ be a new table
3. for $i = 1$ to n :
4. for $j = i$ to n :
5. if $MyCal(i, j) < 0$:
6. $m[i][j] = \infty$
7. else if $i == j$:
8. $m[i][j] = 0$
9. else:
10. $m[i][j] = (MyCal(i, j))^3$
11. $p[0] = 0$

12. for $j = 1$ to n ;

13. $p[j] = \infty$

14. for $i = 1$ to j ;

15. if $p[i-1] + m[i][j] < p[j]$;

16. $p[j] = p[i-1] + m[i][j]$

17. return $p[n]$

算法思路如下:

① 状态转移方程为

$$p[j] = \min_{i=1 \text{ to } j} \{p[i-1] + \text{MyCal}(i, j)\}$$

即第 1 到 j 个单词的最小额外空格的立方和，由第 1 到 i 个单词的最小额外空格的立方和加上 i 到 j 个单词的额外空格决定。

② 根据上式，只需在 j 时，遍历 $i = 1$ 到 j ，就能求出最小。

③ 并且为了简化，令 i 到 j 的单词额外空格数

$$\text{即 } m[i][j] = \begin{cases} \infty & \text{MyCal}(i, j) < 0 \\ 0 & i = j \\ \text{MyCal}(i, j) & \text{其他} \end{cases}$$

复杂度：

空间复杂度为 $O(n^2)$

时间复杂度为 $O(n^2)$