LeetCode712 两个字符串的最小ASCII删除和

> *给定两个字符串`s1, s2`，找到使两个字符串相等所需删除字符的ASCII值的最小和。*

定义：dp[i][j] 表示s1[i]和s2[j]的最小ASCII删除和

显然有以下等式

$$dp[i][j] = \begin{cases} dp[i-1][j-1], & s1 == s2 \\ min(dp[i-1][j] + s1[i], dp[i][j-1] + s2[j]), & s1! = s2 \end{cases}$$

- 注意初始化时，dp大小为[n+1][m+1]，这样dp[0][1]可以理解为s1 =“”, s2 =“%”时情况
- 注意在代码中dp[i][j]对应`s1[i-1]`，`s2[j-1]`

```
int minimumDeleteSum(string s1, string s2) {
        int n = s1.length();
        int m = s2.length();
        vector<vector<int>> dp(n + 1, vector<int>(m + 1));
        dp[0][0] = 0;
        for (int i = 1; i <= n; i++) {
                dp[i][0] = dp[i - 1][0] + (int)s1[i - 1];
        }
        for (int i = 1; i <= m; i++) {
                dp[0][i] = dp[0][i - 1] + (int)s2[i - 1];
        }
        for (int i = 1; i <= n; i++) {
                for (int j = 1; j <= m; j++) {
                        if (s1[i - 1] == s2[j - 1]) {
                                dp[i][j] = dp[i - 1][j - 1];
                        }
                        else {
                                dp[i][j] = min(dp[i - 1][j] + (int)s1[i - 1], dp[i][j - 1]
                        }
                }
        }
        return dp[n][m];
}
```