

所有的区间dp问题枚举时，第一维通常是枚举区间长度，并且一般 $len = 1$ 时用来初始化，枚举从 $len = 2$ 开始；第二维枚举起点 i （右端点 j 自动获得， $j = i + len - 1$ ）

```
for (int len = 1; len <= n; len++) {           // 区间长度
    for (int i = 1; i + len - 1 <= n; i++) { // 枚举起点
        int j = i + len - 1;                  // 区间终点
        if (len == 1) {
            dp[i][j] = 初始值
            continue;
        }

        for (int k = i; k < j; k++) {         // 枚举分割点，构造状态转移方程
            dp[i][j] = min(dp[i][j], dp[i][k] + dp[k + 1][j] + w[i][j]);
        }
    }
}
```

环形dp

我们可以把链延长两倍，变成 $2n$ 个堆，其中 i 和 $i+n$ 是相同的两个堆