

```

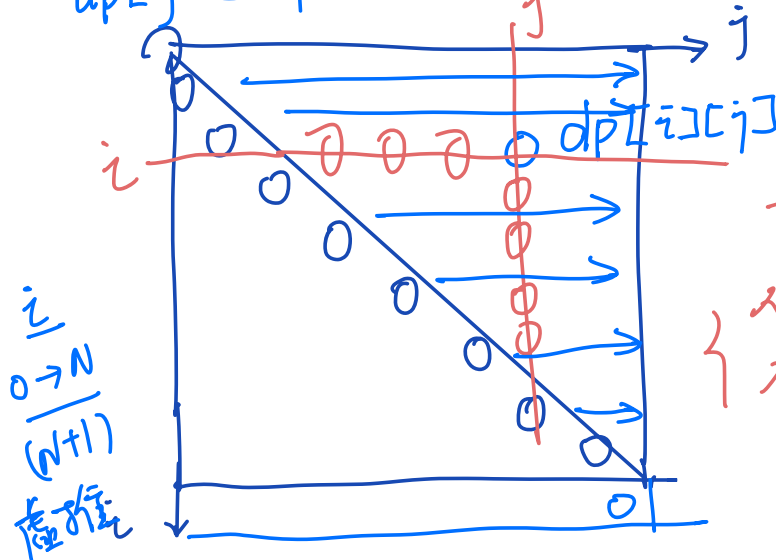
def best(i, j)
    if i > j:
        return 0
    min_cost = +inf
    for r = i to j do:
        cost = best(i, r-1) + best(r+1, j) + w(i, j)
        min_cost = min(min_cost, cost)
    return min_cost

```

best(arr,  $\frac{i}{\Delta}, \frac{j}{\Delta}$ )  
 2维变化

- ① dp数组含义  $dp[i][j]$
- ② 初始化
- ③ 递推公式
- ④ 遍历顺序

$\min_{i < r < j} \{ \text{best}(i, r-1) + \text{best}(r+1, j) + w(i, j) \}$   
 $i = j + 1$   
 $j \leftarrow 0 \text{ to } N-1$



下面, 左边  
 从下到上,  
 从左到右  
遍历顺序

```

def handle(w)
  new dp[0...N][0...N-1] with 0
  for j ← 0 to N-1 do:
    | dp[j+1][j] ← a[j]
    for i ← N to 0 do:
      for j ← i to N do:
        min_cost ← +∞
        for r ← i to j do:
          min_cost ← min { dp[i][r-1]
                           + dp[r][j]
                           + w(i,j),
                           min_cost }
        dp[i][j] ← min_cost
  return dp[1][N]

```

for  $k \leftarrow 1$  to  $n-1$  do

for  $i \leftarrow 1$