

1.

$$10^4 < 100 \log n < n \log n < n^4 < 3^n < 9^n < n^n$$

2.

```
res = 1e9
for k in range(1,4):
    res = min(res, m[1,k] + m[k + 1, 4] + p0 * pk * p4)
```

当 $k = 1$ $m[1,4]=52500$, 当 $k = 2$ $m[1,4]=255000$, 当 $k = 3$ $m[1,4]=91500$

∴ 最少乘次数为 52500

3.

使用 Kruskal 算法求解最短路问题，首先先把所有边按照边权从小到大的顺序排列，然后从排列好的边权列表中从最小的开始选，在保证不构成回路的前提下依次选择最小的边权加入最小生成树，如果构成回路则跳过这条边

```
def kruskal(n, edges):
    dsu = DSU(n)
    edges.sort(key=lambda edge: edge[2]) # 按照边权排
    mst_edges = []

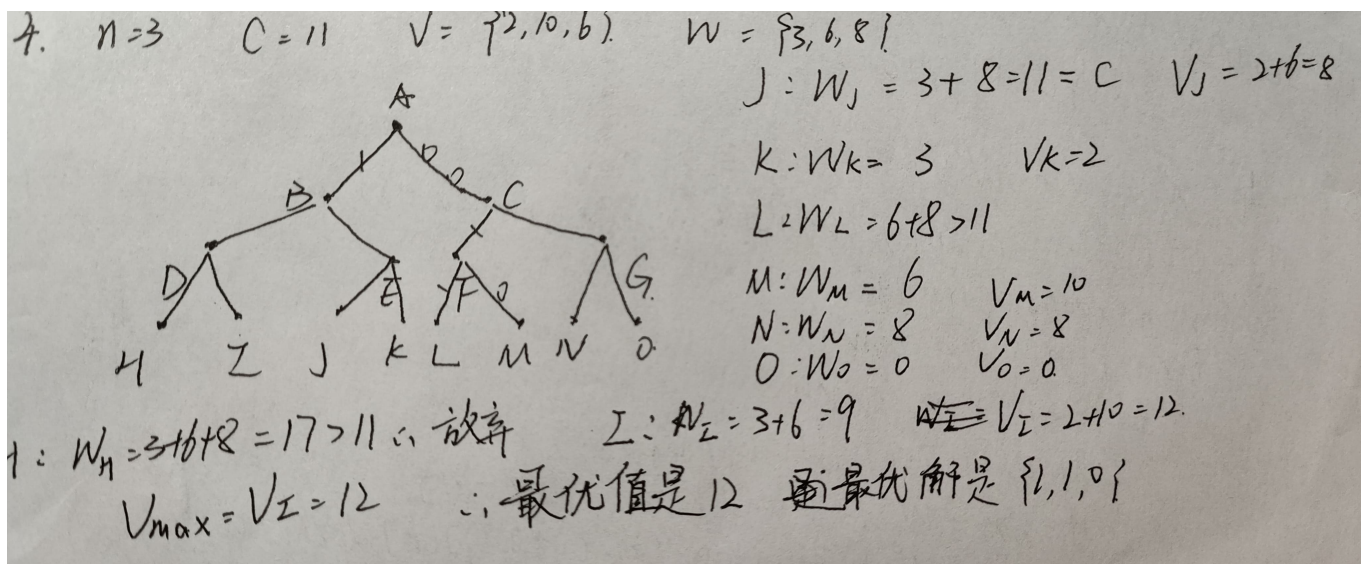
    for u, v, w in edges:
        if dsu.find(u) != dsu.find(v):
            dsu.union(u, v)
            mst_cost += w
            mst_edges.append((u, v, w))

    return mst_cost, mst_edges
```

最终得到最小生成树包括： $A -> B -> C -> D, A -> H -> G, A -> H -> F -> E$

最小生成树的边权和为 $1 + 2 + 3 + 4 + 5 + 5 + 7 = 27$

4.



计算每一个叶子节点的重量和价值，在重量不超过 11 的情况下选取最大的价值

$\therefore$ 最优值是 12，最优解是 $(1, 1, 0)$

5.

```
def find_median(A, B):
    # 先合并，并保证升序
    i = j = 0
    C = []
    while i < len(A) and j < len(B):
        if A[i] < B[j]:
            C.append(A[i])
            i += 1
        else:
            C.append(B[j])
            j += 1
    C.extend(A[i:])
    C.extend(B[j:])

    n = len(C) // 2
    return C[n - 1]
```

6.

$$dp[i][j][k] = \begin{cases} \max(dp[i-1][j][k], dp[i-1][j-w_i][k-c_i] + v_i) & j \geq w_i \text{ 且 } k \geq c_i \\ dp[i-1][j][k] & \text{其他} \end{cases}$$

```

def dp(n, v, w, c, W, V):
    dp = [[[0 for _ in range(V + 1)] for _ in range(W + 1)] for _ in range(n + 1)]
    for i in range(1, n + 1):
        for j in range(0, W + 1):
            for k in range(0, V + 1):
                dp[i][j][k] = dp[i - 1][j][k]
                if j >= w[i - 1] and k >= c[i - 1]:
                    dp[i][j][k] = max(dp[i][j][k], dp[i - 1][j - w[i - 1]][k - c[i - 1]] + v[i - 1])
    return dp[n][W][V]

```