In [3]:
```
import numpy as np
from math import comb
import matplotlib.pyplot as plt

def binomial_tree_option_pricing(S0, K, T, r, u, d, p, N, option_type='call'):
    """
    二叉树模型用于欧式期权定价

    参数:
    - S0: 标的资产当前价格
    - K: 行权价
    - T: 到期时间（年）
    - r: 无风险利率
    - u: 上涨因子
    - d: 下跌因子
    - p: 上涨概率
    - N: 时间步数
    - option_type: 'call' 或 'put'

    返回:
    - 期权在t=0时刻的价格
    """
    dt = T / N
    discount = np.exp(-r * dt)

    # 生成到期日的资产价格
    asset_prices = np.zeros(N + 1)
    for j in range(N + 1):
        asset_prices[j] = S0 * (u ** j) * (d ** (N - j))

    # 初始化到期日的期权价值
    if option_type == 'call':
        option_values = np.maximum(asset_prices - K, 0)
    elif option_type == 'put':
        option_values = np.maximum(K - asset_prices, 0)
    else:
        raise ValueError("期权类型必须是'call'或'put'")

    # 向后递推
    for t in range(N - 1, -1, -1):
        for j in range(t + 1):
            option_values[j] = discount * (p * option_values[j + 1] + (1 - p) * opt

    return option_values[0]

# 示例使用
S0 = 100  # 当前价格
K = 100   # 行权价
T = 1     # 到期时间
r = 0.05  # 无风险利率
u = 1.1   # 上涨因子
d = 0.9   # 下跌因子
p = 0.5   # 上涨概率
N = 3     # 步数

call_price = binomial_tree_option_pricing(S0, K, T, r, u, d, p, N, 'call')
print(f"欧式看涨期权价格: {call_price}")
```

欧式看涨期权价格: 7.110439948142851

In [ ]: