Simulation of variable weight constant liquidity relay, in which stakers can stake assets with custom weightedness, and the contract will maintain the individual ratios of all stakers, using:

1. A self-adjusting global relay weightedness (important for ensuring swaps are executed correctly)
2. Issuance of n-sets of liquidity tokens for an n-asset relay

This simulation uses and ETH:token relay (analagous to ETH:DAI). It is a 2-asset relay, and therefore will use 2 sets of liqiuidity tokens, minted and distributed to stakers.

In [170]:
```
#In this version, liquidity tokens ($_ETH_lt, $_token_lt) are issued direct

e = 2.7182818284590 #Euler's constant, for log. operations

ETH_balance = 1000 #initial ETH balance
ETH_weight = 0.9090909090909091 #normalized weight of ETH
token_balance = 17100 #initial token balance
token_weight = 0.09090909090909091 #normalized weight of tokens

V = ETH_balance^ETH_weight * token_balance^token_weight #formula from Balar
SP = (token_balance/token_weight) / (ETH_balance/ETH_weight) #spot price

ETH_lt = ETH_balance #initial ETH liquidity tokens
token_lt = token_balance #initial token liquidity tokens

println("V: ", V)
println("SP: ", SP)
println("ETH_lt: ", ETH_lt)
println("token_lt: ", token_lt)
```

```
V: 1294.4657245891717
SP: 171.0
ETH_lt: 1000
token_lt: 17100
```

In [171]:
```
#add stake with CUSTOM weighting
#V should change, price (SP) should remain the same
#mint new ETH_lt, token_LT

custom_weight = (0.7,0.3) #custom weights (ETH,token)
ETH_add = 50 #how much ETH to stake
token_add = (ETH_add*SP) * ((1-custom_weight[1])/custom_weight[1]) #how mar
ETH_balance += ETH_add #add stake to ETH pool
token_balance += token_add #add token stake to token pool

ETH_weight = 1 - (1/((ETH_balance * SP / token_balance )+1)) #update ETH we
token_weight = 1/((ETH_balance * SP / token_balance )+1) #update token weig

V = ETH_balance^ETH_weight * token_balance^token_weight #update invariant (

ETH_lt += ETH_add #update ETH liq. token pool
token_lt += token_add #update token liq. token pool

my_ETH_lt = ETH_add #how many liq. tokens go to staker
my_token_lt = token_add #how many liq. tokens go to staker

println("ETH_weight: ", ETH_weight) #updated weight
println("token_weight: ", token_weight) #updated weight
println()
println("ETH liq. tokens: ", my_ETH_lt) #staker's ETH liq. tokens
println("token liq. tokens: ", my_token_lt) #staker's token liq. tokens
println()

#check spot price (should be same)
SP = (token_balance/token_weight) / (ETH_balance/ETH_weight)
println("Spot price: ", SP)
```

```
ETH_weight: 0.8963414634146342
token_weight: 0.10365853658536586

ETH liq. tokens: 50
token liq. tokens: 3664.285714285715

Spot price: 171.0
```

In [172]:
```
#add another stake with CUSTOM weighting (same as above, just a new stake)
#V should change, price should remain the same
#mint new ETH_lt, token_LT

custom_weight = (0.4, 0.6) #custom weights (ETH,token)
ETH_add = 120 #how much ETH to stake
token_add = (ETH_add*SP) * ((1-custom_weight[1])/custom_weight[1])
ETH_balance += ETH_add #update ETH liquidity pool
token_balance += token_add #update token liquidity pool

ETH_weight = 1 - (1/((ETH_balance * SP / token_balance )+1)) #update ETH we
token_weight = 1/((ETH_balance * SP / token_balance )+1) #update token weig

V = ETH_balance^ETH_weight * token_balance^token_weight #update invariant

ETH_lt += ETH_add
token_lt += token_add

your_ETH_lt = ETH_add #new staker's ETH liq. tokens
your_token_lt = token_add #staker's token liq. tokens

println("ETH_weight: ", ETH_weight)
println("token_weight: ", token_weight)
println()
println("ETH liq. tokens: ", your_ETH_lt)
println("token liq. tokens: ", your_token_lt)
println()

#check spot price (should not change from staking)
SP = (token_balance/token_weight) / (ETH_balance/ETH_weight)
println("Spot price: ", SP)
```

```
ETH_weight: 0.7951456310679612
token_weight: 0.20485436893203882

ETH liq. tokens: 120
token liq. tokens: 30779.999999999996

Spot price: 171.0
```

In [173]:
```
#Verify that original staker still has originally-weighted assets and balar
show(my_ETH_lt/ETH_lt * ETH_balance == 50)
println()
show(my_token_lt/token_lt * token_balance == 3664.285714285715)
println()
show(round((my_token_lt/token_lt * token_balance) / ((my_ETH_lt/ETH_lt * ET
    round(0.3/0.7, digits = 5))
```

```
true
true
true
```

In [174]:
```
#add another stake with CUSTOM weighting
#V should change, price should remain the same
#mint new ETH_lt, token_LT

custom_weight = (0.2, 0.8) #custom weights (ETH,token)
ETH_add = 30 #how much ETH to stake
token_add = (ETH_add*SP) * ((1-custom_weight[1])/custom_weight[1])
ETH_balance += ETH_add #update ETH liquidity pool
token_balance += token_add #update token liquidity pool

ETH_weight = 1 - (1/((ETH_balance * SP / token_balance )+1)) #update ETH we
token_weight = 1/((ETH_balance * SP / token_balance )+1) #update token weig

V = ETH_balance^ETH_weight * token_balance^token_weight #update invariant

ETH_lt += ETH_add
token_lt += token_add

her_ETH_lt = ETH_add #new staker's ETH liq. tokens
her_token_lt = token_add #new staker's token liq. tokens

println("ETH_weight: ", ETH_weight)
println("token_weight: ", token_weight)
println()
println("ETH liq. tokens: ", your_ETH_lt)
println("token liq. tokens: ", your_token_lt)
println()

#check spot price (should not change from staking)
SP = (token_balance/token_weight) / (ETH_balance/ETH_weight)
println("Spot price: ", SP)
```

```
ETH_weight: 0.7400881057268722
token_weight: 0.2599118942731278

ETH liq. tokens: 120
token liq. tokens: 30779.999999999996

Spot price: 170.99999999999997
```

In [186]:
```
#Verify that second staker still has originally-weighted assets and balance
show(your_ETH_lt/ETH_lt * ETH_balance == 120)
println()
show(your_token_lt/token_lt * token_balance == 30779.999999999996)
println()
show(round(((your_ETH_lt/ETH_lt * ETH_balance)*SP) / (your_token_lt/token_l
    round(0.4/0.6, digits = 5))
```

```
true
true
true
```

In [187]:
```
#Does it work with swaps?

#sell tokens (buy ETH)

V = ETH_balance^ETH_weight * token_balance^token_weight #update invariant (

sale = 5000 #how many tokens to send to contract (swap for ETH)
token_balance += sale #add sale tokens to contract
old_ETH_balance = ETH_balance #store for calculation below
ETH_balance = e^((log(V / (token_balance^token_weight)))/ETH_weight) #updat
ETH_received = old_ETH_balance - ETH_balance #ETH received for swapped toke
println("Tokens sold: ", sale)
println("ETH received: ", ETH_received)
println()

V = ETH_balance^ETH_weight * token_balance^token_weight #update invariant (
SP = (token_balance/token_weight) / (ETH_balance/ETH_weight)
println("V: ", V)
println("SP: ", SP)
```

```
Tokens sold: 5000
ETH received: 27.93961048900792

V: 3478.921626843823
SP: 187.22353395727927
```

In [195]:
```
#Verify that original staker still has originally-weighted assets and balar
#balances will have changed b/c of swap, but weithed ratio should remain co
show(round((my_token_lt/token_lt * token_balance) / ((my_ETH_lt/ETH_lt * ET
    round(0.3/0.7, digits = 5))

println()

#Verify that second staker still has originally-weighted assets and balance
#balances will have changed b/c of swap, but weithed ratio should remain co
show(round(((your_ETH_lt/ETH_lt * ETH_balance)*SP) / (your_token_lt/token_l
    round(0.4/0.6, digits = 5))

println()

#Verify that third staker still has originally-weighted assets and balances
#balances will have changed b/c of swap, but weithed ratio should remain co
show(round(((her_ETH_lt/ETH_lt * ETH_balance)*SP) / (her_token_lt/token_lt
    round(0.2/0.8, digits = 5))
```

```
true
true
true
```

In [196]:
```
#sell ETH (buy tokens)
V = ETH_balance^ETH_weight * token_balance^token_weight #update invariant (

sale = 5 #how much ETH sent to contract (swapped for tokens)
old_token_balance = token_balance #store for calculation below
ETH_balance += sale
token_balance = e^((log(V / (ETH_balance^ETH_weight)))/token_weight) #updat
tokens_received = old_token_balance - token_balance
println("ETH sold: ", sale)
println("tokens received: ", tokens_received)
println()

V = ETH_balance^ETH_weight * token_balance^token_weight
SP = (token_balance/token_weight) / (ETH_balance/ETH_weight)
println("V: ", V)
println("SP: ", SP)
```

```
ETH sold: 5
tokens received: 928.4879426185798

V: 3478.921626843653
SP: 184.1821031389585
```

In [197]:
```
#Verify that original staker still has originally-weighted assets and balar
#balances will have changed b/c of swap, but weithed ratio should remain co
show(round((my_token_lt/token_lt * token_balance) / ((my_ETH_lt/ETH_lt * ET
    round(0.3/0.7, digits = 5))

println()

#Verify that second staker still has originally-weighted assets and balance
#balances will have changed b/c of swap, but weithed ratio should remain co
show(round(((your_ETH_lt/ETH_lt * ETH_balance)*SP) / (your_token_lt/token_l
    round(0.4/0.6, digits = 5))

println()

#Verify that third staker still has originally-weighted assets and balances
#balances will have changed b/c of swap, but weithed ratio should remain co
show(round(((her_ETH_lt/ETH_lt * ETH_balance)*SP) / (her_token_lt/token_lt
    round(0.2/0.8, digits = 5))
```

```
true
true
true
```