AA33 reverted (or OOG)

OOG = Out Of Gas

ThevalidatePaymasterUserOp function of the paymaster contract either reverted or ran out of gas.

Possible solutions

- Verify that theverificationGasLimit
- is high enough to cover thevalidatePaymasterUserOp
- function's gas costs.
- If you are using your own paymaster contract, verify that thevalidatePaymasterUserOp
- function is implemented with the correct logic, and that the user operation is supposed to be valid.
- If you are using your own paymaster contract, make sure the signature verification logic of the paymaster is implemented correctly. Namely, correct implementations shouldnot
- revert when the paymaster signature is invalid. If you are not looking to use neither time-based signatures, you should simply returnuint(1)
- for invalid signatures in thevalidatePaymasterUserOp
- function.
- If you are using a paymaster service like Pimlico, and the user operation is well formed with a high enoughverificationGasLimit
- [.reach out to them](#)
- .
- If you arenot
- looking to use a paymaster to cover the gas fees, verify that thepaymasterAndData
- field is not set.

```

constuserOperationHash=awaitbundlerClient.sendUserOperation({
paymasterAndData:"0x3b912be0270b59143985cc5c6aab452d99e2b4bb0000000000000000000000000000000000000000000000000000000000000000064c09574000000000000000000000000000000000000000000(
paymasterAndData:"0x", ...,// the rest of the user operation })

```

- If all else fails, investigate why thevalidatePaymasterUserOp
- function reverted using tools like[Tenderly](#)
- .