# 4.https://stackoverflow.com/questions/73681122/state-overrides-overflowerror-python-int-too-large-to-convert-to-c-ssize-t

**T:**State Overrides: OverflowError: Python int too large to convert to C ssize\_t

**Q:**I've written some simple state override functions in a contract that I'd like to use to query specific data about a contract. The idea is that we call a series of view functions and aggregate the outputs "on chain" before returning the output all at once  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
However, I'm seeing the following error. Any idea where I'm going wrong?  
  
OverflowError: Python int too large to convert to C ssize\_t  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Contract code  
  
// SPDX-License-Identifier: MITpragma solidity ^0.8.9;import "@openzeppelin/contracts/token/ERC721/ERC721.sol";contract NftSnapshot { function snapshotUsers(address \_nftAddress, uint256[] calldata \_tokenIds) public view returns (address[] memory) { address[] memory owners = new address[](\_tokenIds.length); for (uint256 i; i < \_tokenIds.length; i++) { owners[i] = IERC721(\_nftAddress).ownerOf(\_tokenIds[i]); } return owners; } function snapshotTokenMetadata(address \_nftAddress, uint256[] calldata \_tokenIds) public view returns (string[] memory) { string[] memory tokens = new string[](\_tokenIds.length); for (uint256 i; i < \_tokenIds.length; i++) { tokens[i] = IERC721Metadata(\_nftAddress).tokenURI(\_tokenIds[i]); } return tokens; }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Python code  
  
from typing import List, Optionalfrom web3 import Web3from eth\_abi import decode\_abiimport timedef virtual\_contract\_call( web3\_interface: Web3, abi: str, runtime\_bytecode: str, fn\_name: str, fn\_args: List, latest\_block: Optional[int] = None,) -> List: virtual\_contract\_address = "0x37F5eCc33e530D495C336866d7471FF9ad1C2469" contract = web3.eth.contract(address=virtual\_contract\_address, abi=abi) function\_call\_payload = contract.encodeABI(fn\_name, fn\_args) state\_override = {virtual\_contract\_address: {"code": runtime\_bytecode}} block\_number = web3\_interface.eth.getBlock( "latest" if not latest\_block else latest\_block ).number res = web3.eth.call( {"to": virtual\_contract\_address, "data": function\_call\_payload}, block\_number, state\_override=state\_override, ) return resweb3 = Web3(Web3.HTTPProvider("https://eth-mainnet.g.alchemy.com/v2/ZpQFH9fKeo5GUpbViaz\_VkhyqaWJUfhF"))abi = """[ { "inputs":[ { "internalType":"address", "name":"\_nftAddress", "type":"address" }, { "internalType":"uint256[]", "name":"\_tokenIds", "type":"uint256[]" } ], "name":"snapshotTokenMetadata", "outputs":[ { "internalType":"string[]", "name":"", "type":"string[]" } ], "stateMutability":"view", "type":"function" }, { "inputs":[ { "internalType":"address", "name":"\_nftAddress", "type":"address" }, { "internalType":"uint256[]", "name":"\_tokenIds", "type":"uint256[]" } ], "name":"snapshotUsers", "outputs":[ { "internalType":"address[]", "name":"", "type":"address[]" } ], "stateMutability":"view", "type":"function" }]"""runtime\_bytecode = ""res = virtual\_contract\_call( web3\_interface=web3, abi=abi, runtime\_bytecode=runtime\_bytecode, fn\_name="snapshotUsers", fn\_args=[Web3.toChecksumAddress("0x60e4d786628fea6478f785a6d7e704777c86a7c6"), [\*range(1, 10000)]], #MAYC NFT contract)snapshot = decode\_abi(["address[]"], bytes.fromhex(res.hex()[2:]))print(snapshot)  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Here's a quick repo I spun up if you'd like to get deeper into the scaffolding  
  
https://github.com/chumbalayaa/state-override-functions  
  
Full backtrace  
  
Traceback (most recent call last): File "<stdin>", line 1, in <module> File "<string>", line 102, in <module> File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/eth\_abi/codec.py", line 181, in decode\_abi return decoder(stream) File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/eth\_abi/decoding.py", line 127, in \_\_call\_\_ return self.decode(stream) File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/eth\_utils/functional.py", line 45, in inner return callback(fn(\*args, \*\*kwargs)) File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/eth\_abi/decoding.py", line 173, in decode yield decoder(stream) File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/eth\_abi/decoding.py", line 127, in \_\_call\_\_ return self.decode(stream) File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/eth\_abi/decoding.py", line 144, in decode stream.push\_frame(start\_pos) File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/eth\_abi/decoding.py", line 95, in push\_frame self.seek\_in\_frame(0) File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/eth\_abi/decoding.py", line 84, in seek\_in\_frame self.seek(self.\_total\_offset + pos, \*args, \*\*kwargs)OverflowError: Python int too large to convert to C ssize\_t  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

2 **Answer**

**A1:**The integer is too large for the Python compiler to compile it down to C code.There are various methods to determine the max size / max value of a Python int.  
  
Python 2.x: sys.maxintPython 3.x: sys.maxsize  
  
If you really need a number that long, you should check out NumPy Data Types such as numpy.ulonglong for storing large numbers.

**C1:**Can you point me to which number is too large?

**C2:**I digged a bit through the repository of eth\_abi to better understand the traceback. But I cannot really put my finger on the exact problem. Maybe try shortening you runtime\_bytecode or other variables that might cause this to narrow down the problem. I sadly cannot run solcjs on my pc atm, otherwise I would try to use a debugger to see the variables and how big the integers are

**A2:**the int is too large, thats what it says