

Python Code

```
import torch
m1 = torch.randn(40, 40, 40).cuda()
model = torch.nn.AdaptiveAvgPool2d(
    output_size = [1, 67108607]).cuda()
model(m1)
```

CUDA Code

```
// Host Code
at::native::adaptive_avg_pool2d_out_cuda_template
int64_t osizeH = output_size[0];
int64_t osizeW = output_size[1];
output.resize_({sizeD, osizeH, osizeW});
adaptive_average_pool<<<...>>>(..., osizeH, osizeW, ...);

// Device Code at::native::adaptive_average_pool
__global__ void adaptive_average_pool(..., scalar_t *output,
                                      int osizeH, int osizeW, ...)
{ for(oh = ostartH; oh < oendH; oh += ostepH) {
    for(ow = ostartW; ow < oendW; ow += ostepW) {
        // Potential out-of-bounds calculation
        scalar_t *ptr_output = output + oh*osizeW + ow;
        // Accessing invalid memory
        *ptr_output = sum / kH / kW;
```