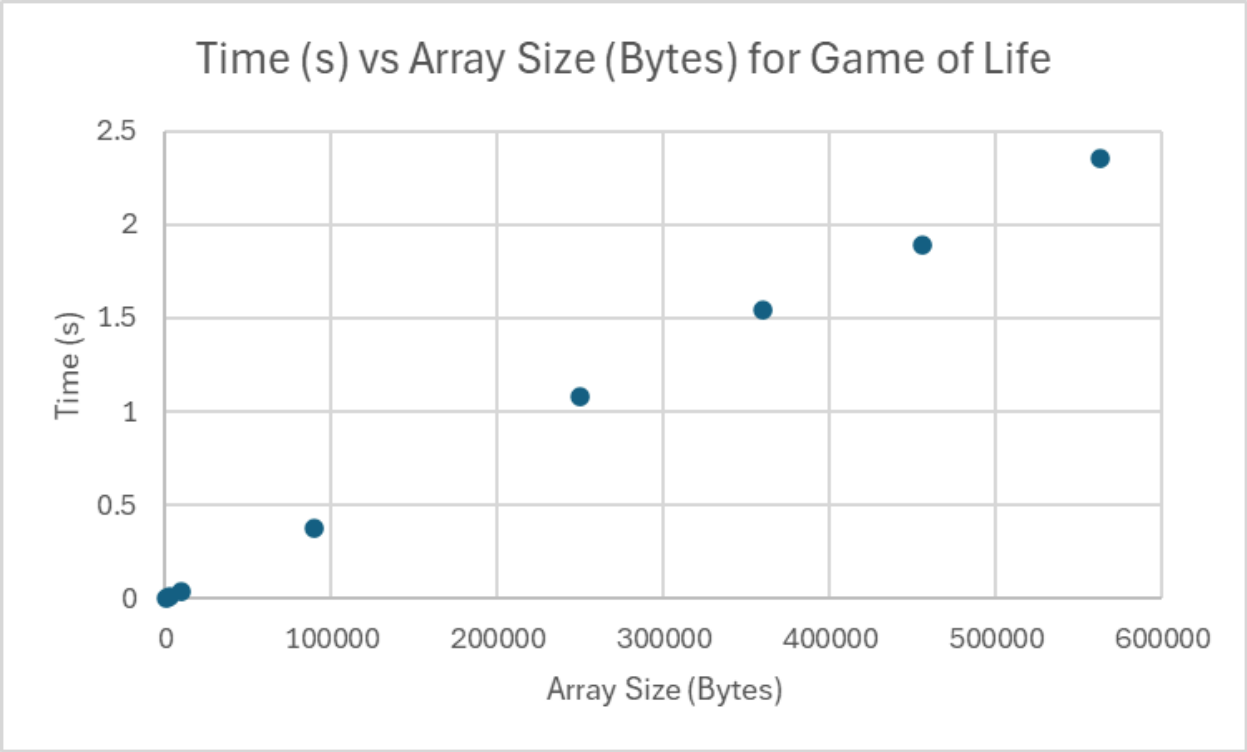
To reach above a second of runtime for 80*24, I need around ~14000 generations. I am running the program on a kali vm on our school laptop.

To reach above a second of runtime for a 1000*1000 grid, I need around ~23 generations.

If we have 500 generations, I need around 230*230 grid to reach a second of runtime.

Here is my plot of the runtime. As illustrated, the time increases linearly with array size. I did not use any optimization flags in the compilation for this analysis. I did try some for fun and it does make a significant difference, often improving runtime by over .1 seconds.



Here are the results for lscpu. The L1 cache and L2 cache is 128 Kb and 2 Mb respectively. The CPU has four cores.

```
None
Architecture:                    x86_64
  CPU op-mode(s):                32-bit, 64-bit
  Address sizes:                 40 bits physical, 48 bits virtual
  Byte Order:                    Little Endian
CPU(s):                          4
  On-line CPU(s) list:           0-3
Vendor ID:                       AuthenticAMD
```

```
  Model name:                AMD Ryzen 7 PRO 6850U with Radeon Graphics
    CPU family:              25
    Model:                   68
    Thread(s) per core:      1
    Core(s) per socket:      1
    Socket(s):               4
    Stepping:                1
    BogoMIPS:                5389.70
    Flags:                   fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 clflush mmx fxsr
                              sse sse2 syscall nx mmxext rdtscp lm constant_tsc
rep_good nopl tsc_reliable nonstop_ts
                             c cpuid extd_apicid tsc_known_freq pni pclmulqdq
ssse3 fma cx16 sse4_1 sse4_2 movbe popc
                             nt aes xsave avx hypervisor lahf_lm abm sse4a
misalignsse 3dnowprefetch osvw vmmcall ara
                             t overflow_recov succor
Virtualization features:
  Hypervisor vendor:         VMware
  Virtualization type:       full
Caches (sum of all):
  L1d:                       128 KiB (4 instances)
  L1i:                       128 KiB (4 instances)
  L2:                        2 MiB (4 instances)
  L3:                        64 MiB (4 instances)
NUMA:
  NUMA node(s):              1
  NUMA node0 CPU(s):         0-3
Vulnerabilities:
  Gather data sampling:      Not affected
  Indirect target selection: Not affected
  Itlb multihit:             Not affected
  L1tf:                      Not affected
  Mds:                       Not affected
  Meltdown:                  Not affected
  Mmio stale data:           Not affected
  Reg file data sampling:    Not affected
  Retbleed:                  Not affected
  Spec rstack overflow:      Vulnerable: Safe RET, no microcode
  Spec store bypass:         Vulnerable
  Spectre v1:                Mitigation; usercopy/swapgs barriers and __user
pointer sanitization
  Spectre v2:                Mitigation; Retpolines; STIBP disabled; RSB
filling; PBRSB-eIBRS Not affected; BHI Not a
```

```
                            ffected
  Srbds:                    Not affected
  Tsa:                      Vulnerable: Clear CPU buffers attempted, no
microcode
  Tsx async abort:          Not affected
```