# Garbage Collection

- What is garbage collection?
- Reference Counting
- Mark and Sweep
- Stop and Copy

# What is garbage collection?

- Memory Management
- Recall:
  - C -> user allocated and deallocated memory manually
  - Java, Ocaml, etc. -> memory that is not being used is freed by a garbage collector

But how do we create a “good” garbage collector?

| Memory | Don’t Free | Free |
|---|---|---|
| In use | Good | Bad |
| Not in use | Fine | Good |

# Reference Counting

- Keep track of how many pointers (references) there are to a certain piece of memory
- Free memory when it hits 0 references
- Example

Pros? Cons?

# Reference Counting Pros and Cons

Pros:

- Fast
- Can be done while program is running

Cons:

- Cyclic Data
- Uses extra memory for keeping track of refs

# Mark and Sweep

- Stop program, mark what is reachable from the stack
- Sweep away (free) unreachable memory
- Example

Pros? Cons?

# Mark and Sweep Pros and Cons

Pros:

- Solves issue of cyclic data
- Memory efficient

Cons:

- Fragmentation (draw)
- Program stops

# Stop and Copy

- Divide memory into 2 halves, alive and dead
- Go through stack, copy over all reachable data to "dead" partition
- Swap partitions and continue

- Example

Pros? Cons?

# Stop and Copy Pros and Cons

Pros:

- Solves issue of fragmentation
- Still no issue of cyclic data

Cons:

- Halves usable memory
- Stops program