Garbage collection

The Basics

Major concerns

- Explicit memory management
  - Reclaiming objects at the right time
- Garbage collection
  - Discriminating live objects from garbage
- Both
  - Fast allocation
  - Fast reclamation
  - Low fragmentation

Automated memory management

- Runtime system automatically
  - Detects dead objects (garbage detection)
  - Reclaims dead objects (garbage reclamation)
  - Garbage collection
- Preserves software development time
  - Relieves programmer burden
  - Less prone to errors
- Utilized by most modern OOP and scripting languages
  - Python, Java, C#, php

Runtime system performs GC

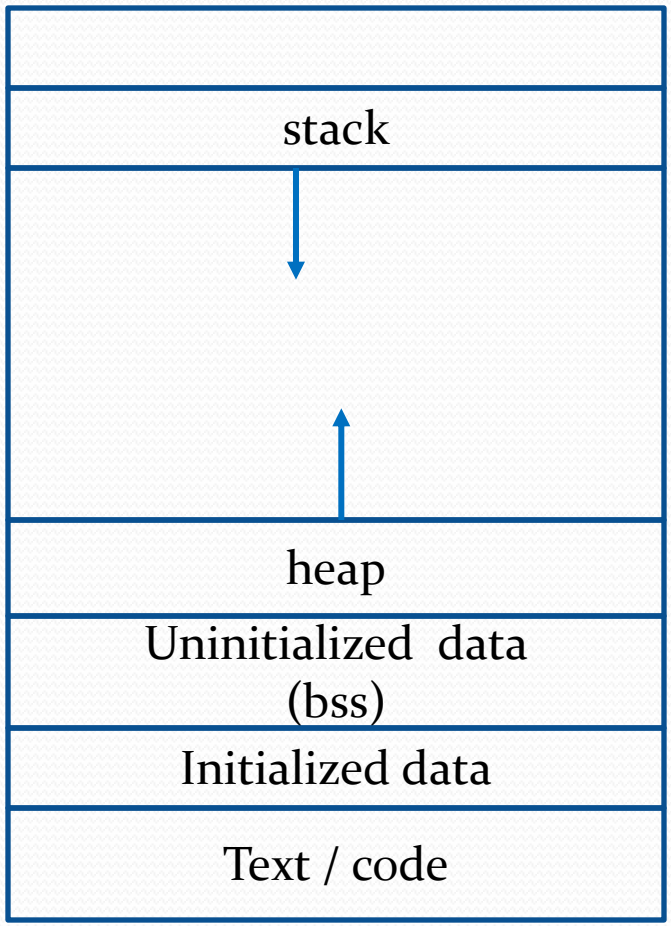
- E.g. Java virtual machine (JVM)
  - Software execution engine that executes your Java programs
  - Java interpreter that converts byte code into OS specific commands
  - Handles related tasks
    - Memory management (GC implemented in JVM)
    - Security
    - Multithreading

Major concerns

- Explicit memory management
  - Reclaiming objects at the right time
- Garbage collection
  - Discriminating live objects from garbage
- Both
  - Fast allocation
  - Fast reclamation
  - Low fragmentation

Layout of a program in memory

High address



Low address

Command line args and environment variables

Initialized to 0 by exec

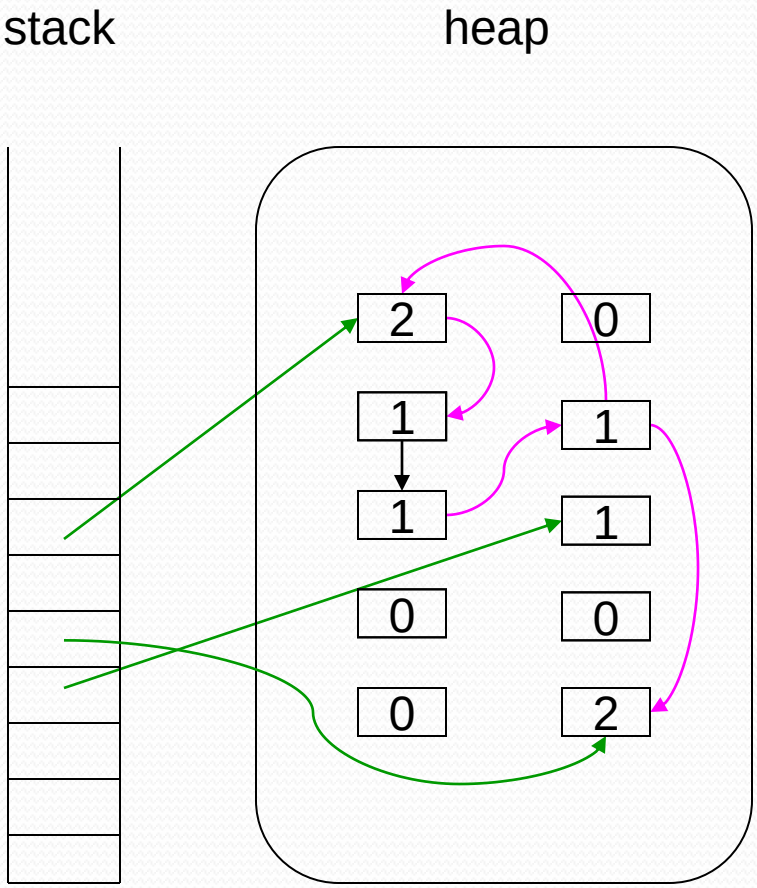
Read from program file by exec

Determining object liveness

- Live objects are needed in the computation
  - Now or in the future
- Prove that an object is not live (dead) and reclaim its storage
- Reclaim dead objects soon, after it is last used
- How do we estimate *liveness* in practice?
  - Approximate *liveness* by reachability from outside the heap
    - Unreachable objects are garbage (reclaim storage)
    - Reachable objects are live and must not be reclaimed

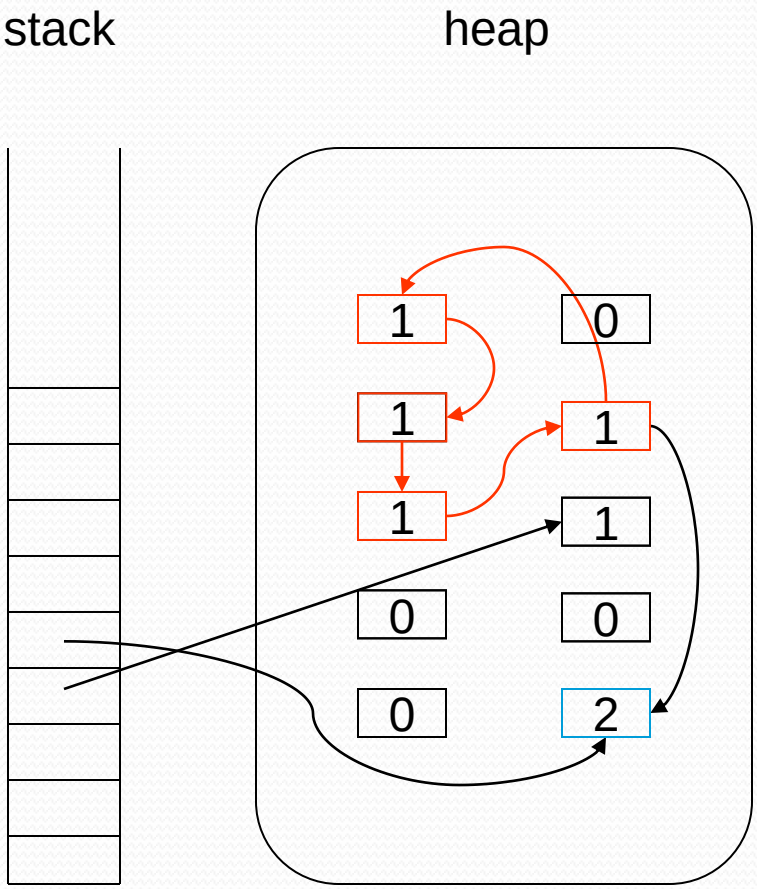
Identifying garbage

- **reference counting (reachability)**
- An integer is associated with every object, summing
  - Stack references
  - Heap references
- Objects with reference count of zero are dead



Problems with reference counting

- Standard problem is that objects in cycles (and those touched by such objects) cannot be collected (reclaimed)
- Overhead of counting can be high



Identifying garbage

- **Tracing (reachability)**
- Trace reachability from **root set**
  - Processor registers
  - Program stack
  - Global variables
- Objects traced are reachable
- All other objects are unreachable (garbage)

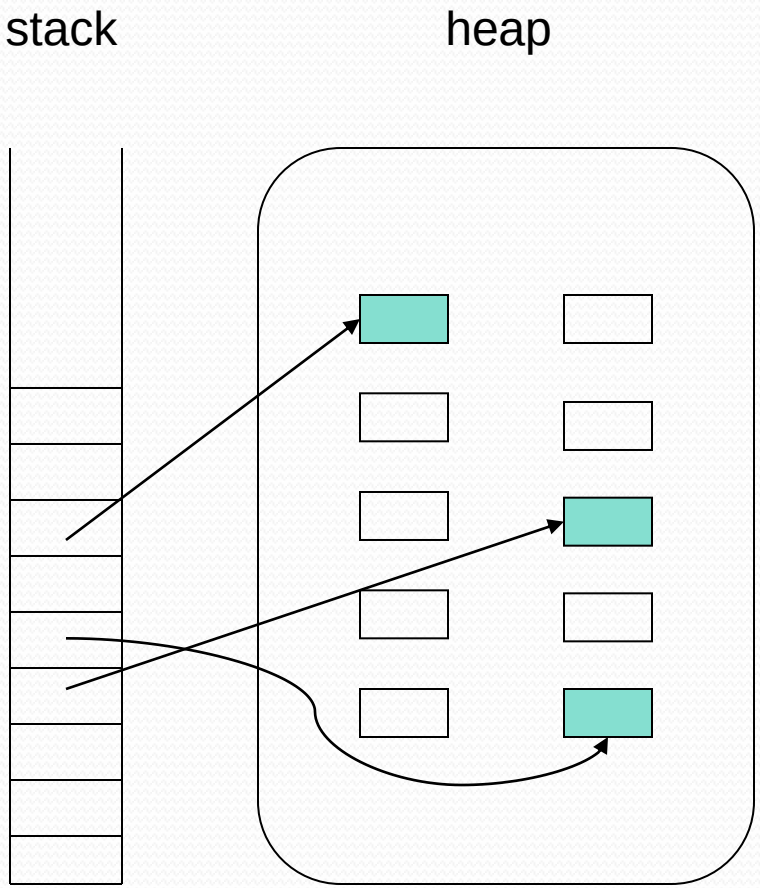
The marking phase

- To find the dead objects, use the process of *calculatus eliminatus*
  - Find all live objects
  - All others are dead

The marking phase

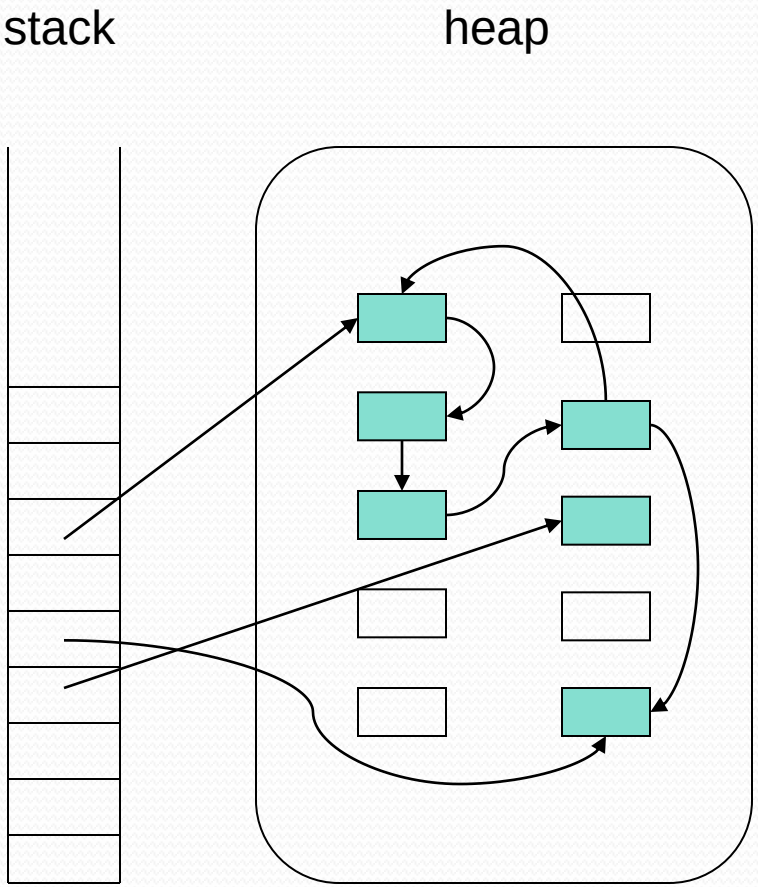
- To discover the dead objects, we
  - Find live objects

- Pointers from the stack to the heap make objects live



The marking phase

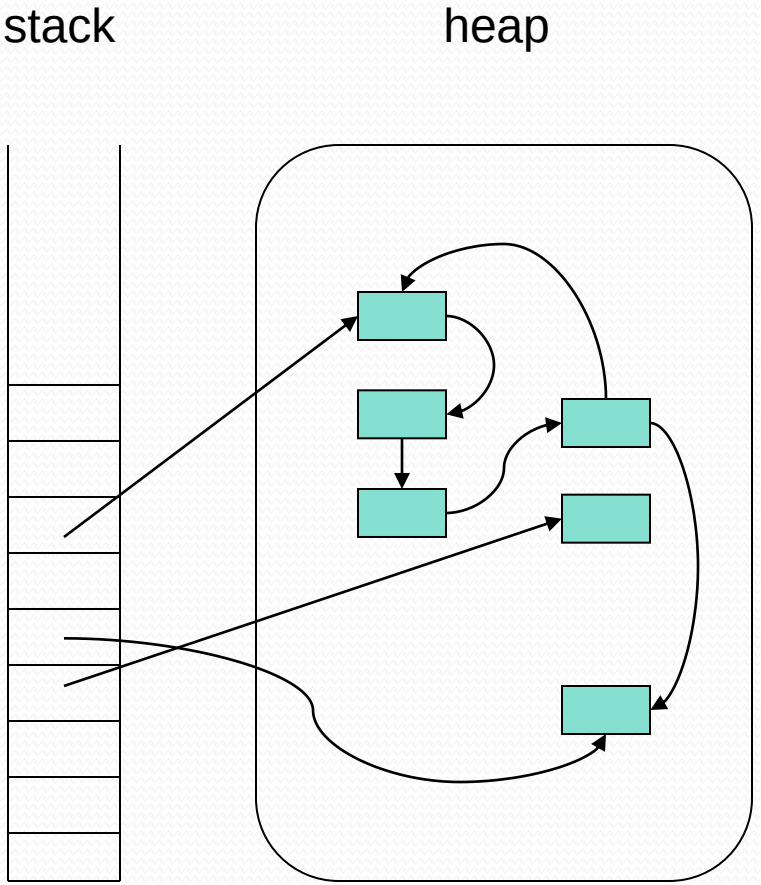
- To discover the dead objects, we
  - Find live objects



- Pointers from the stack to the heap make objects live
- These objects make other objects live

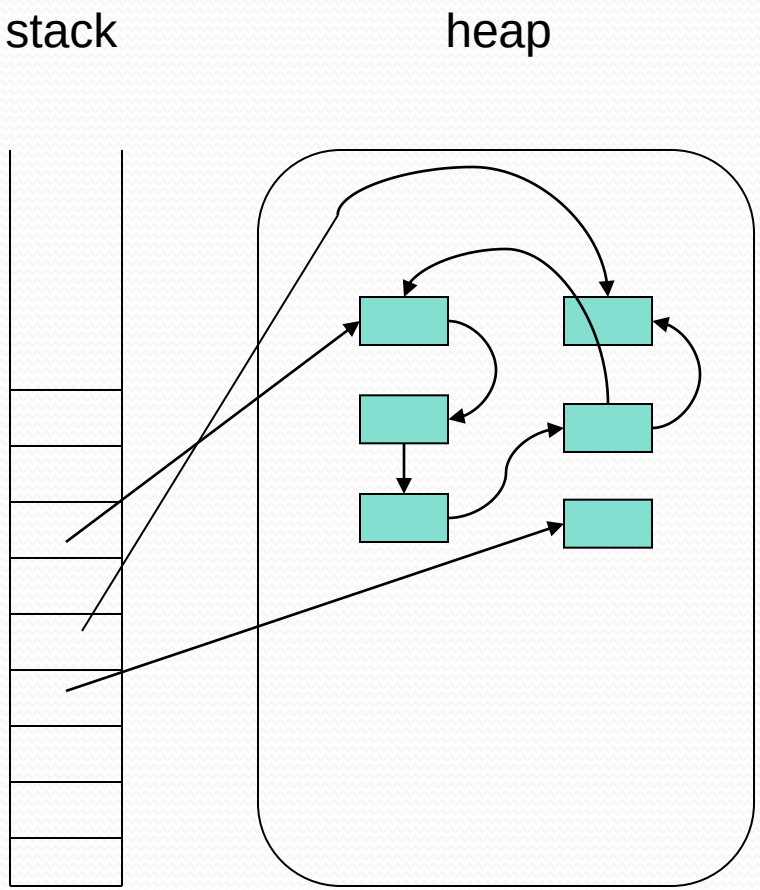
The sweep phase

- To discover the dead objects, we
  - Find live objects
  - Sweep all others away as dead



Mark and sweep: Tracing example

- To discover the dead objects, we
  - Find live objects
  - Sweep all others away as dead
  - Perhaps compact the heap
  - Problem:
    - Mark phase can take unbounded time



Garbage collection design choices

- Stop-the-world
- Incrementality
- Hybrid
- Concurrency
- Parallelism

Stop-the-world collectors

- Typically used on uniprocessor systems
- Suspend application
- Run collector from start to finish
- Resume application

Stop-the-world collectors

- Execution costs?
  - Pause time
  - Discovery of live objects (how long does it take?)
  - Instruction overhead (per instruction)
  - Delay between object death and collection
  - Number of collectible objects collected
  - Overall execution time
  - Worst-case vs average case performance
  - frequency

Incremental collection

- Interleave GC with application
- Note: for full heap tracing
  - Pause time increases with heap size
- Incremental tracing
  - Bounded tracing time
  - Conservative assumption
    - All other objects in heap are live
  - Remember pointers from objects in heap
    - Add such pointers to root set for tracing