Automatic Memory Management

Storage Allocation

- Static Allocation
  - Bind names at compile time
  - Pros:
    - Fast (no run time allocation, no indirection)
    - Safety : memory requirements known in advance
  - Cons:
    - Sizes must be known at compile time
    - Data structures can't be dynamically allocated
    - No recursion

Storage Allocation

- Stack Allocation
  - activation records (frames)
  - push + pop on proc entrance / exit
  - Implications:
    - Recursion is possible
    - Size of local data structures may vary
    - Stack allocated local names can't persist
    - Can only return objects of statically known size
    - Enables function pointers (no nesting though)

Storage Allocation

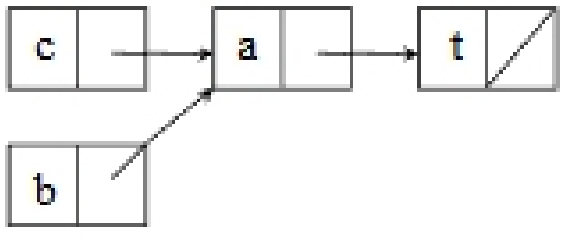
- Heap Allocation
  - Alloc and dealloc from a heap in any order
  - Advantages:
    - Local data structures can outlive procedure
    - Facilitates varying sized recursive data structures
    - Can return dynamically sized objects
    - Closures (function + environment)

Reachability

- What can a program manipulate directly?
  - Globals
  - Locals (in registers, on stack, etc)
  - In C, random locations
- Root nodes
- Live nodes - pointer reachability

Problems w/ Manual Allocation

- Garbage - "unreachable" but not free
- Dangling references
- Sharing



- Failures
  - Invalid accesses, out of memory errors, etc...