Garbage Collection

Lecture topics
I. Intro to GC
II. Reference Counting
III. Mark-Sweep GC

I. Intro to GC

Storage organization

- Stack: function activation records
- Heap: objects, arrays, strings

Heap allocation example

r = {{i=1, n=3)};
r.n = {{i=2, n=1)};
s = {{i=3, n=2)};
a = r, s;
s = null;
r.n = null;
s = {{i=4, n=1)};

Garbage = heap object that will not be used in the future (e.g., (i=2, n=1) is unreachable)

Garbage collection (GC) = reclaim memory of garbage, for reuse in later allocation

Various techniques:

- Manual
- Garbage collection
- Reference counting
- Mark-sweep GC
- Copying GC

Tack leaves memory management unspecified.

II. Reference Counting

Keep implicit 'count' field in every heap object.

- When creating pointer: increment count
- When overwriting pointer: decrement count
- When count drops to 0:
  - Recursively decrement counts
  - Free object

Problem: reference counting does not reclaim cycle

III. Mark-Sweep GC

Traverse objects from roots to find reachable
Reclaim all unreached objects

Use mark stack (reached) must see pointer
mark bit in each heap object

States of object:

- Free
- Unreached
- Not on mark stack
- Mark bit is 0
- Scanned
- Not on mark stack
- Mark bit is 1
- Unscanned
- On mark stack
- Mark bit is 1

Example:

Mark P, A, C, E
Shake A, A, A, C, E

[Diagram of mark-sweep GC process]