

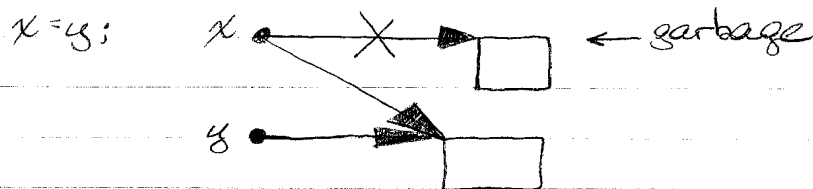
316 11-12-09



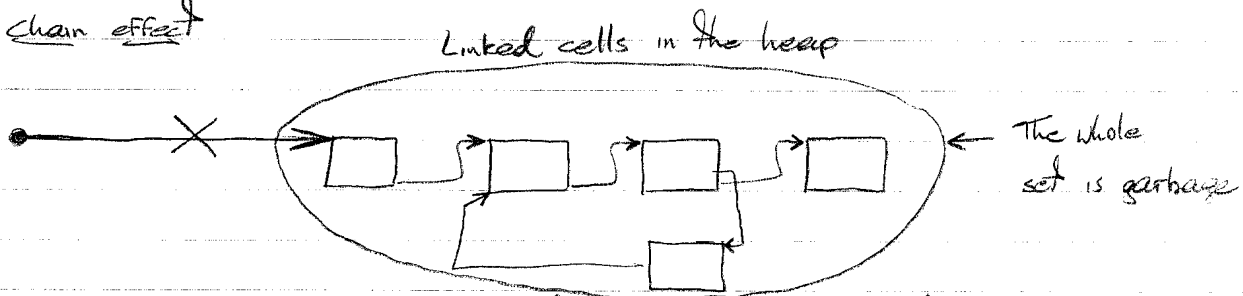
Garbage Collection

A memory-cell graph is a graph of nodes and directed links:

- 2 kinds of nodes:
 - root nodes: static vars' cells, activation records' cells, registers, which contain references to heap cells
 - non-root nodes: all allocated cells in the heap
- The directed links are references from/to nodes as defined above.
- A non-root node is reachable if it can be reached by a chain of one or more references from one of the root nodes. A non-root node is a garbage node if it isn't reachable.
- Main scenarios that may create garbage nodes:
 - assignment between references



- chain effect

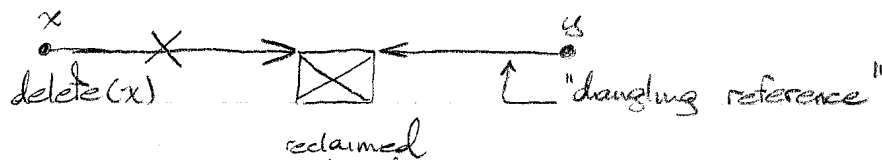


- At any time during program execution, the heap divides into:
 - allocated nodes
 - reachable nodes
 - garbage nodes
 - free segments

Explicit Destructors vs Garbage Collection

Destructors

- Erroneous application
- apply destructor to a cell that is still referenced from elsewhere



- forget to apply destructor just before the referenced cell becomes garbage
 - creates garbage cells never to be reclaimed "memory leak" problem
- No runtime overhead of garbage collector
 - In single-cpu computers, the garbage collector pauses the program's execution
- mark-and-swap method:
 - all allocated nodes will be maintained by some data structure (let's assume a linked list).
 - every allocated cell in the heap has a boolean mark field.
 - a global boolean variable "USED" shall be utilized. At any time, the mark field of any newly allocated cell will be assigned the current value of USED.
 - a graph traversal algorithm is needed.
- garbage collection algorithm:

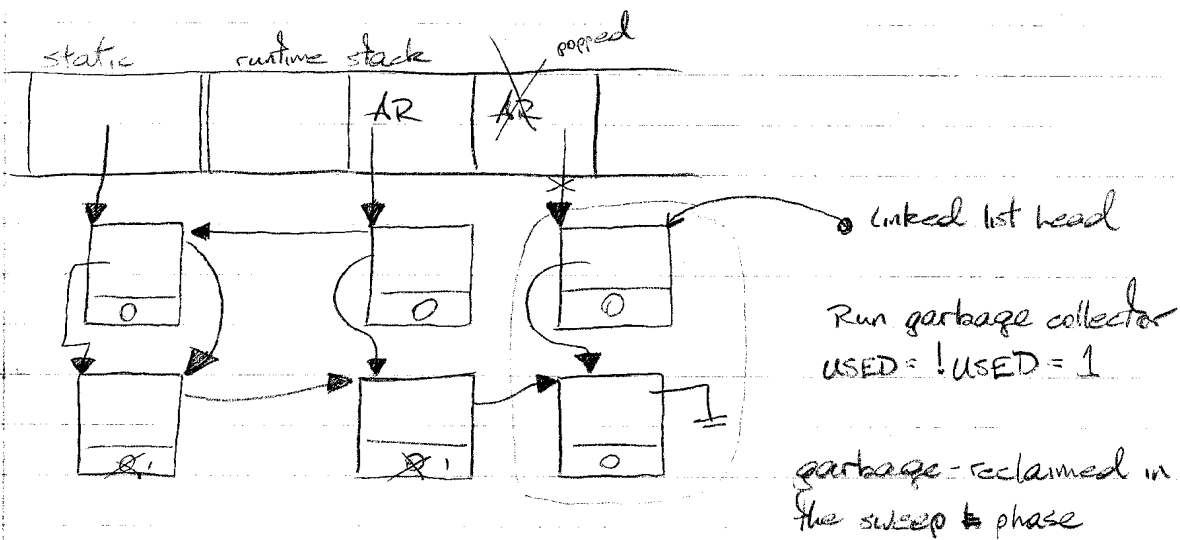
Mark Phase

- $USED = !USED$ //negate bool value
- run the graph traversal algorithm starting from each of the root nodes, and set the mark-fields of the visited nodes to the current value of USED. //garbage nodes' mark fields have previous value of USED.

Sweep Phase

- Traverse the linked-list of the allocated nodes and ~~reclaim~~ the nodes whose mark-fields values are distinct from the current value of USED to the free segment.

Suppose current value of USED = 0



Stop and copy Method

- A graph traversal algorithm is used
- The entire heap is divided into ACTIVE and UNUSED areas of equal size
- All newly allocated cells will be allocated in ACTIVE area only.

Garbage collector Algorithm

- run the graph traversal algorithm from each of the root nodes and copy (relocate) all visited cells to a contiguous portion of UNUSED area
- switch ACTIVE and UNUSED areas.