

Lec 28: Garbage Collection

May 1, 2026

Heapify (Wrapup)

How do we build a heap?

- Seems easy: call **Insert()** on each item one at a time, takes $O(n \log n)$ total time
 - This was the method used when heaps were originally invented
- But: it's possible to build a heap in $O(n)$ total time!
- This subroutine is called Heappify

Heapify: Idea



- **Idea:** go *bottom-up* through the heap, making sure that taller and taller subtrees satisfy the heap property
- How do we make sure the leaves satisfy the heap property
 - Don't need to do anything—they have no children
- What about the parents of the leaves?
 - Sift down once

Heapify



- Go through all items of the heap bottom up (from back to front in the array). Call sift down on all of them.
- Let's do an example on the board
- Running time?
- The time to call sift-down on a node is the height of its subtree (the longest path to any leaf)
 - $n/2$ leaves have 0 swaps
 - $n/4$ nodes have ≤ 1 swap
 - $n/8$ nodes have ≤ 2 swaps
 - $n/16$ nodes have ≤ 3 swaps
- With a little math, the total # swaps is $< 2n$.

HeapSort

- One of the most efficient sorting algorithms
- First, Heapify in $O(n)$
- Then, `RemoveMin()` every element, placing it in the correct (next) position in the new array
 - $O(\log n)$ time per `RemoveMin()`
 - $O(n \log n)$ time overall

Garbage Collection

Getting Rid of Items

- The remove methods we've seen have often skipped an intuitive step
- We “disconnect” the item, but we don't *really* delete it
- Let's look at `LinkedList.java` for an example
- Does this mean the old Node just sits there, taking up space?

Some Memory Experiments

- The Two Towers lab generates 2^n subsets
- On the board: how much space do we anticipate this would take if $n = 26$?
- Let's look at what actually happens when we run the code. We'll use the htop utility to monitor RAM usage
- Now, let's modify the code a little bit to *explicitly store* all subsets, and run the code again.
- TwoTowers.java takes very little space. TwoTowersSpace.java takes $\approx 10\text{GB}$. What is happening?

Garbage Collector



- Java has a process, called the Garbage Collector, which deletes objects on your behalf
- When we were done with a subset, Java deleted it for us (so space was small)
- Let's talk about this in more detail:
 - What does it mean to be “done” with an object?
 - How does Java figure this out?

Finishing With an Object

- In Java, we keep track of objects using [references](#)
 - `Student s1 = new Student()` means that `s1` stores a reference to the new `Student` object
- If there is no reference to an item, we can't access it—so we're safe to delete it
- In `TwoTowers.java`, each time the for loop iterates, we no longer have a reference to the old subset. It's deleted.
- In `TwoTowersSpace.java`, we *keep* references to all subsets in an array. None are deleted!

How Garbage Collection Works



- Java has an *entire process* devoted to doing the following; it runs at all times. It is called the “mark and sweep” algorithm.
 - Java looks at all memory references of (long story short) local variables. It marks them as being *in use*
 - Then it looks at all of *their* references; they are also marked
 - So on and so forth
 - Then, it scans through all available memory and deletes all unmarked objects
 - Their memory can be used by other objects—when you call new, it might overwrite this newly-freed memory

What Garbage Collection Means for Us

- What happens when we delete a `LinkedList` node?
 - It is **disconnected** from the list
 - This (probably) means there are no references to it; it will be deleted by the garbage collector later on
- Same for tree nodes, etc.
- It is not possible in Java to delete an object/free memory manually. You just delete all references to the object and wait for it to be cleaned up by the garbage collector
- Python also uses a garbage collector; it works largely the same way

Garbage Collection Tradeoffs

- Some languages (famously, C) do not have a garbage collector. *You* must manually delete anything you want deleted.
- **In pairs:** What are some advantages of having a garbage collector? What are some disadvantages?