

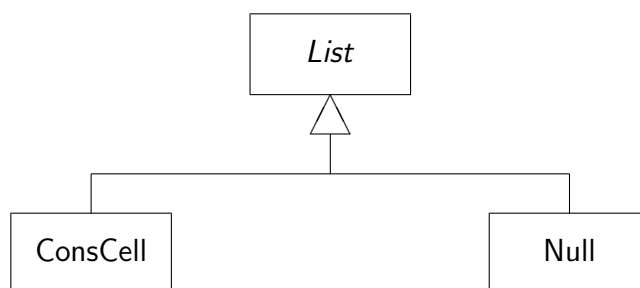
# Homework # 7

## due Tuesday, March 15, 3:30 PM

### 1 Introduction to Squeak

This assignment requires using Squeak. You can use `weise/miller` from an X11 terminal (but not remotely through putty) or, better, you can download Squeak onto your home machine. To get the recommended “Squeak By Example” image working, you need to download a VM for your machine and also the `SqueakV39.sources` sources file. You will also need the “Squeak By Example” image and changes files. For simplicity, you can put all four files in a single folder on your home machine. On my MacBook, the VM complains about not being ready for prime-time, but it seems to work fine with the SBE image.

In the MPL textbook, it discusses implementing ML style lists in Java using static methods and a ‘nil’ value. It then uses a facade object `IntList` to avoid static methods. The resulting code is still rather procedural, rather than object-oriented. In this Homework, we will use a more OO solution, and implement a destructive merge sort of a list. Unlike the textbook, we have a `Null` class for empty lists and put both classes under a class `List` which is not intended to have instances:



You will do Exercise 13.8 (page 236): destructive `sortMe` for both cons cells and null objects. The sort will have the following steps:

1. If the list has only one element, return itself.
2. Otherwise, destructively split itself into two, and recursively sort each half.
3. Then merge the results destructively (using the `<` or `<=` messages).

The split and merging are destructive, in that we re-direct cons cell pointers rather than allocate any new ones. Both the `splitMe` and `mergeMeWith:` methods should be implemented recursively. The code is actually very short.

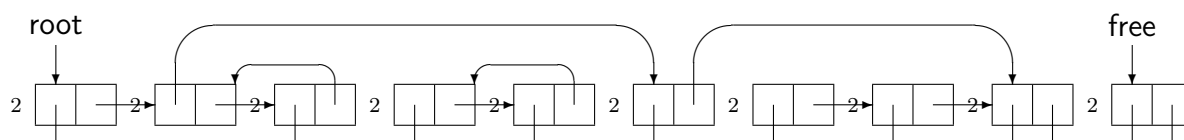
We provide a skeleton file of the solution, `Homework-7.st.SKEL`, in the public directory `$CLASSHOME/src/homework7/`. Copy this file into the directory where your squeak image is, and rename it has `Homework-7.st`. Use the File Browser to “file in” the file. Use the System Browser to ‘file out’ the Homework-7 category and put the resulting `Homework-7.st` file in your AFS directory for Homework #7.

This file also includes a test suite of seven tests. Follow the directions on pages 160–161 of SBE to run the tests.

## 2 Garbage Collection

For the next homework, you will implement a mark-sweep collector (see MPL, page 265) starting with a single root pointer **root**. Assume (along with the textbook) that each block allocated has to keep an extra word for the length of the block in the word before the pointer returned by the allocator.

Suppose we have an allocator managing a heap of exactly 30 words. Each memory word in the heap can hold either an integer or a pointer, and the allocator can tell which is which. Suppose further that after some allocation and changes, the memory heap is as follows:



Now, show the process that the allocator must go through in order to perform mark-sweep garbage collection using the single root (shown). The allocator will have an auxiliary array of bits (size 30) for marks. Write all the steps the allocator does in order to complete a mark-sweep garbage collection on this example:

1. It's marking from pointer 1; it checks to see if mark 1 is set. It isn't, and so it gets the size (2), and marks words 0,1 and 2.
  2. Then it (recursively) marks using the contents of 1 and 2 (as seen below)
  3. The contents of 1 is the null pointer. Nothing more to do.
  4. The contents of 2 is the pointer 4 (location 3 has the "2" in it). It checks to see if mark 4 is set. It isn't, ...
- ⋮

Please continue all the mark steps and then describe the sweep steps.

Finally draw what the heap looks like after the garbage collection process is done: make sure to indicate what the free list is.

## 3 Submitting Your Work

Put the changed `Homework-7.st` file in your `homework7` directory.