

# Python Garbage Collector

## Summary

Create a Python program (`hw3.py`) that simulates a Mark-Sweep garbage collection algorithm. In this program, named pointers are referred to using variable names such as `p`, `stackPtr`, `temp3`, etc. Heap blocks are referred to using integers.

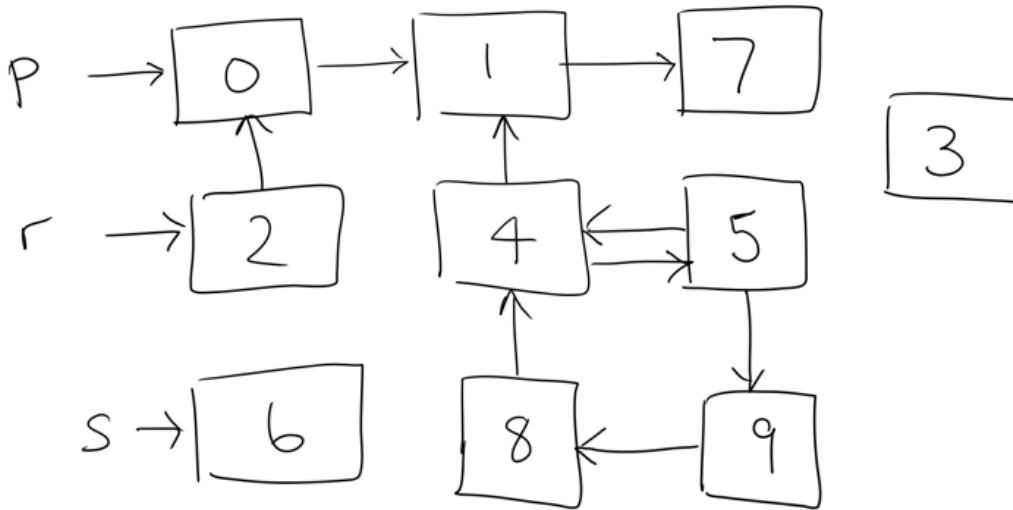
The program has the following functionality:

1. Get the name of an input file from the command line (using `sys.argv`). WARNING: Do not prompt the user for a file name.
2. Process the file. The first line will contain  $n$  number of heap blocks--the heap blocks will be identified using the numbers 0 through  $n - 1$ . Each subsequent line will contain an ordered pair either in the form :
  - *named pointer, heap block* (example: `p,10` means `p` points to heap block 10)
  - *heap block, heap block* (example: `7,3` means heap block 7 points to heap block 3)
3. Perform the mark-sweep algorithm.
4. Output which heap blocks are marked and which heap blocks should be reclaimed (swept).
5. The program must contain a function named `mark_sweep` that takes a filename string and returns a dictionary with two string keys 'marked' and 'swept' where the values for the 'marked' key is the ordered list of integers which are the marked nodes you'll print out and 'swept' is the ordered list of integers which are the swept nodes you'll print out. You can assume I'll run the following doctest (where *example.csv* is the example shown above):

```
>>> mark_sweep('example.csv')
{'marked': [0, 1, 2, 6, 7], 'swept': [3, 4, 5, 8, 9]}
```

## Example

Diagram of heap:



Corresponding input file:

```

10
p,0
0,1
1,7
r,2
2,0
4,1
4,5
5,4
5,9
s,6
8,4
9,8

```

Output from program for this example:

```

Marked nodes: 0 1 2 6 7
Swept nodes: 3 4 5 8 9

```

## Notes

- You may assume the input is valid and properly formatted. Note: there are no spaces in an input file line.
- A variable name consists of letters, digits, and underscores but cannot begin with a digit. (Remember that the program will only be tested with valid variable names.)
- You may assume the variable names are unique (i.e., a variable can only point to one heap block).
- You may assume there is at least one heap block, i.e., the number on the first line will be greater than zero.
- The output must print the marked nodes and swept nodes in numerical order separated by spaces.
- Your algorithm must run in polynomial time but does not need to be optimal.

