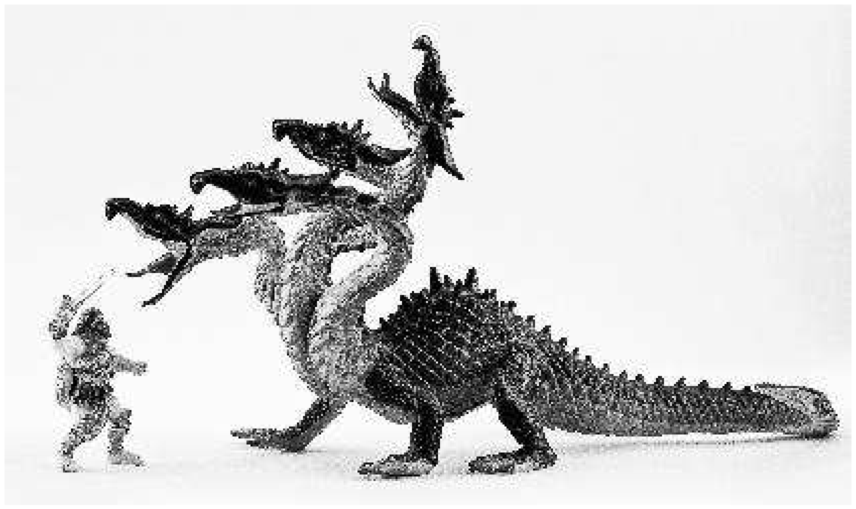
Computer Science I — CSCI 141
Hercules and the Hydra — Homework

11/06/2013

1 Problem



The second of Hercules' twelve Labors was to kill the Hydra of Lerna. The Hydra was a serpent-like creature with many heads. What made this tricky for Hercules was that each time he cut off a head from the Hydra, two more grew back in its place.

You will write a program, **hydra.py**, that will simulate Hercules' battle against the Hydra.

1.1 Rules

The rules of the battle are as follows:

- The battle will consist of rounds. If Hercules has not defeated the Hydra within 100 rounds, he will give up.
- In each round of the battle, Hercules severs one of the heads from the Hydra.
- Assume that the size of the severed head is represented by the variable `headsize`.
  - If `headsize == 1`, nothing grows back in its place.
  - If `headsize > 1`, two heads of size `headsize // 2` grow back in its place.
- After every k^{th} round of the battle, the remaining Hydra heads (including any that just sprouted!) grow in size by 1.

Your code must allow Hercules to follow one of two strategies in his fight against the Hydra. In the first strategy, Hercules will always sever the smallest remaining head of the Hydra in each round. In the second strategy, Hercules will always sever the largest remaining head of the Hydra in each round.

1.2 Input

Your program will read as input the name of a file containing the details of the Hydra as well as Hercules' strategy. The input file specification as is follows:

- The first line of the file will contain a single word indicating Hercules' strategy. The word will be either `largest` or `smallest`.
- The second line of the file will contain a sequence of positive integers, separated by spaces. Each number represents the size of a Hydra head at the beginning of the battle.
- The third and final line of the file contains a single positive integer that indicates the repeating period - the number of rounds - after which the remaining Hydra heads grow in size by 1. (See the last rule in the above section.)

An example input file, `hyrda1.txt`, is as follows:

```
smallest
8 7 3
10
```

You may assume that the input file is correctly formatted. To read just one line at a time from a file, you can use the `readline` function as in the following:

```
fp = open('hydra1.txt')
line = fp.readline()
```

1.3 Code Reuse

You will find it helpful to use the source code provided in this week's lecture notes. Import `array_heap.py` into your program to have access to the heap functions described in class.

1.4 Program Operation

When run, your program should prompt for input of the name of a file containing the battle information as described above.

Running your program should result in output something like the following:

```
Please input name of file to read:  hydra1.txt
Hercules' strategy:  smallest
Initial Hydra heads: [8, 7, 3]
Hydra growth period: 10

Result:  Hercules slays the Hydra after 29 rounds.
```

If Hercules is unable to defeat the Hydra within 100 rounds, the final line of your output should state:

```
Result:  Hercules gives up after 100 rounds.
```

1.5 Requirements

In order to receive full credit, you must use a heap data structure to represent the battle. When you create your heap, one of the arguments is the maximum size to which the heap may grow. You may use a worst case estimate of the size of the heap at the end of the battle: 100 + number of heads Hydra has at the beginning of the battle.

1.6 Testing

In addition to the example input and output described above, the following two sample outputs are provided so that you may test your program. You will need to copy the input data into your own test file in the appropriate format.

- Output:
  ```
  Please input name of file to read: hydra2.txt
  Hercules' strategy:  largest
  Initial Hydra heads:  [2, 2, 2, 1]
  Hydra growth period:  7

  Hercules slays the hydra after 20 rounds
  ```
- Output:
  ```
  Please input name of file to read: hydra3.txt
  Hercules' strategy:  largest
  Initial Hydra heads:  [2, 2, 2, 2]
  Hydra growth period:  8

  Hercules gives up after 100 rounds
  ```

1.7 Grading

- 20%: The program reads the input file correctly and generates appropriate output messages.
- 70%: The program uses a heap data structure and correctly determines the outcome of the battle.
- 10%: The submitted file follows the course style recommendations.

1.8 Submission

Submit your solution as a file called `hydra.py` to the MyCourses dropbox for this assignment.