

Week 12

Hash Wrap up, Exam review, Python tricks
Arnamoy Bhattacharyya

Outline

- Hashing wrap up
- Exam tips
- Question simulation

Hashing Wrap up

Performance of our hashtable as compared to python dict

Other use cases of hashing

1. Passwords are stored as hashed values
2. When you forget password, hard to retrieve even for the admin
 - a. Same hash value → two different original passwords
3. During log in → hash values are matched for given and stored
4. Odds are low for two passwords to have same hash value
5. Hackers still follow brute force approach

Exam logistics

The final exam

When and where:

<http://www.artsci.utoronto.ca/current/exams/apr18>

CSC148H1S	A - K	MON 30 APR	PM 2:00 - 5:00	EX 100	
CSC148H1S	L - WA	MON 30 APR	PM 2:00 - 5:00	EX 200	
CSC148H1S	WI - ZHAN	MON 30 APR	PM 2:00 - 5:00	EX 300	
CSC148H1S	ZHAO - ZZ	MON 30 APR	PM 2:00 - 5:00	EX 310	

- Check that you know the room and how to get to it, in advance!
 - www.osm.utoronto.ca/map/
- Plan to be there 10-15 minutes before .. Exam starts on the hour!

Preview

PLEASE HAND IN

UNIVERSITY OF TORONTO
Faculty of Arts and Science

April 2017 Examinations

CSC148H1S
Duration — 3 hours
No aids allowed.

PLEASE HAND IN

Student Number: | | | | | | | | | | | | | | | |

Last Name: _____

First Name: _____

Do not turn this page until you have received the signal to start.
(In the meantime, please fill out the identification section above,
and read the instructions below.)

This exam consists of 7 questions on 22 pages (including this one).
When you receive the signal to start, please make sure that your copy
of the exam is complete.

Please answer questions in the space provided. If you need additional
space, clearly indicate on the question page where to find your answer.

You will earn 20% for any question you leave blank or write "I cannot
answer this question" on. You may earn substantial part marks for
writing down the outline of a solution and indicating which steps are
missing.

You must achieve 40% of the marks on this final exam to pass this
course.

There is a Python API at the end of this exam.

1: ____/ 8

2: ____/ 8

3: ____/10

4: ____/ 6

5: ____/ 6

6: ____/10

7: ____/ 8

TOTAL: ____/56

Good Luck!

Exam materials

object-oriented programming and design: lecture slides and example code, in-class exercises, weeks 1–2, lab #1, lab #2, and assignment #1

abstract data types, stacks, queues: lecture slides and example code, week 3, lab #3

linked lists: lecture slides and example code, in-class exercise, week 4, lab #4

modularity, functional programming idiom lecture slides and example code, week 5

reading, writing recursion on nested lists: lecture slides and example code, in-class exercise, week 6, lab #5

Exam materials

recursion on general trees: lecture slides and example code
week 7, in-class exercises on contains and leaf,
lab #6

recursion on binary trees: lecture slides and example code week
8, lab #7

binary search trees, insertion, deletion, mutation: lecture slides
and example code week #9, in-class exercise

efficiency, recursion, recursive structures: lecture slides and
example code week #10, lab #8

big-Oh, big-Theta, hash table: lecture slides and example code
week #11

hash table, tracing and traps: lecture slides and example code
week #12

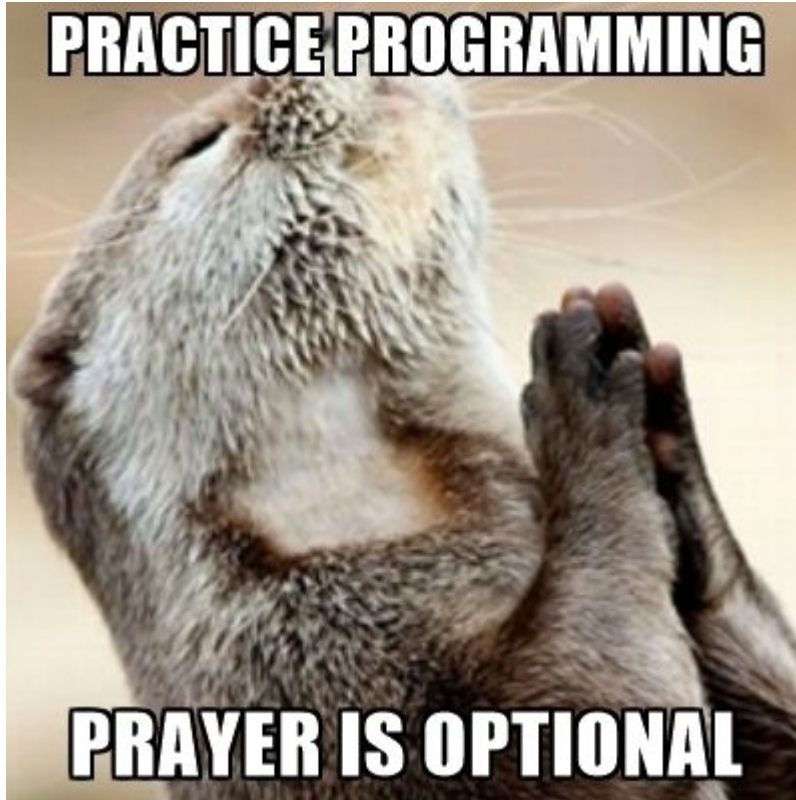
General tips: How to study

- I read all the slides and examples, I'm ready! ❌
- I did the labs at the time, I must be ready now! ❌
- I did all the examples myself from scratch, without looking at the solutions and I got it right!
plus
- I did extra examples and looked into the documentation when I was stuck, then solved the extra work! ✅

More tips on how to study

- Work on a lab handout, or incomplete code, before looking at the solution
- Practice some more examples
- Study groups can challenge each other, critique solutions
- Use office hours!
- Did I mention practice?

True fact ...



More General Tips

- Do not panic! Take a deep breath, you've got this!
- This is your chance to show us what you've learned
- We WANT to give you the credit that you've earned
- Read carefully!
- What is the question asking?
- If there's anything unclear, please ask!
- Keep track of your time
- Some questions take more time than others
- Do not spend too much time on a question if you are stuck – might want to revisit it later

Question simulation

8 Marks out of 54 ..

```
def swap_even(t, depth=0):
    """ Swap left and right children of nodes at even depth.
    ... # stuff omitted for space...
    >>> b1 = BinaryTree(1, BinaryTree(2, BinaryTree(3)))
    >>> b4 = BinaryTree(4, BinaryTree(5), b1)
    >>> print(b4)
        1
          2
            3
4
    5
    >>> swap_even(b4)
    >>> print(b4)
        5
4
    1
          3
            2
    """
```