YSC2229: Assignment 3

Deadline: March 31, 2021 at 11.59pm

(Strict Deadline: No Late Submission Accepted)

Submit your work in the jupyter-notebook format via Canvas by the deadline. Include your non-code answers on your submitted jupyter notebook. This is an individual assignment. No collaboration with anyone is allowed. **Violation on this academic integrity is a serious offence and will be reported to the College's academic integrity committee**.

Grading criteria: clarity of your explanation/discussion, correctness of the code, runtime (ensure the algorithm in the code is the most efficient one), user-friendliness of the code (ensure people who read your code easily understand by writing remarks for every function, having consistent naming of variables and functions, and keeping the code simple), bug free, etc. Coding tips: `https://www.aversan.com/coding-standards-and-best-practices-2/`

1. Write code in python to implement the randomized solution of the vertex cover problem. The input of you function is a graph consisting of nodes and edges, represented by an open hash table. The output is the list of nodes and the total number of nodes. For the code, you must follow this algorithm:

   APPROX-VERTEX-COVER(G)
   1 $C = \emptyset$
   2 $E' = G.E$
   3 **while** $E' \neq \emptyset$
   4 let (u, v) be an arbitrary edge of E'
   5 $C = C \cup \{u, v\}$
   6 remove from E' every edge incident on either u or v
   7 **return** C

   Discuss your code on your jupyter-notebook. The users must be informed on how they use/run your code.

2. Write code in python to implement sorting in linear time using radix sort and counting sort. The input is a set of random words, e.g., < dog, cat, rain, umbrella, digit >. The output is the set sorted in descending order, e.g, < umbrella, rain, dog, digit, cat>. You must write your own sorting functions. Explain your algorithm and code on your jupyter-notebook. The users must be informed on how they use/run your code.

3. Write code in python to implement a queue using two stacks, where each of the stack is implemented using a singly linked list. The queue operations must include: ENQUEUE and DEQUEUE. You must write your own stack operation functions and linked list functions. Explain your algorithm and code on your jupyter-notebook. The users must be informed on how they use/run your code.

4. Write code in python to implement the open hash table that is under the simple uniform hashing assumption (SUHA) and robust to adversarial attacks on the efficiency of the table. The operations include: insert, delete, and search (by all attributes). The table should be scalable to a large number of data of a hospital's patients, consisting: name, date of birth, IC number, and illness. Show empirically that your implemented table is under the SUHA. Explain your algorithm, parameters, code and results on your jupyter-notebook. The users must be informed on how they use/run your code.