CSC 403
Haonan(Harry) Chen
Homework 3 (written)

3.13

If choices are between ordered array and unordered linked list, I would choose ordered array structure to implement the symbol table when 10^3 put() and 10^6 get() operations are needed. In this case, get() operations are 100 times more than put() operations, we should choose the structure which cost less in get(). Ordered array using binary search which will cost $\ln(N)$in get(), although this implementation behave very slow for insertion which will cost 2N in worst case, but on average it will cost N, it depends on how those two operations intermixed. So the total cost is less then $2*10^3*\ln(10^6)$ by using ordered array, compare to unordered linked list which will cost less than 10^3*10^6, we definitely choose order array.

3.14

In the case of 10^6 put() and 10^3 get() operations, I still choose to use ordered array. The main reason is unordered linked list runs very slow for large STs like this one, although it will keep the run time of insert at constant N. Let's compute the total cost for both of method, for unordered linked list, it will cost $10^6 * 10^3 = 1000*10^6$ in the worst-case. For ordered array, it will cost $2*10^6 *\ln(10^3) \approx 21* 10^6$ in worst-case. Comparing the coefficient of those two, Apparently the ordered array the choice.

3.2.2

1. A,C,E,H,R,S,X
2. X,S,R,H,E,C,A
3. A,X,C,S,E,H,R
4. A,X,C,E,S,H,R
5. A,C,E,X,S,R,H

3.2.4

The answer is d, since after 2,7,3, the rest keys are all in the range of (3,7), however 8 is not included, so 8 is not examined.