DSA
PS1, due Wed 21 8:00 PM

See notes for review of sliding window algorithm.

1. Read up through Chapter 5 (skip Chapter 2) of the textbook.

2. First duplicate. Given a string *S* return the index of the first repeated character. *S* = "abca" returns 3, since the first duplicate is "a" at index 3.

3. Longest increasing subsequence of an array. Given an array *L* return the length of the longest *increasing* subsequence. *L* = [1,2,3,2,5] --> return 3, as [1,2,3] longer than [2,5]; [1,1,1,1,1] --> return 1, as [1] is the longest increasing subsequence; [1,0,1,0,1,0] --> return 2, as len([0,1]) is 2.

4. Longest substring containing unique characters. Given a string *S* return the length of the longest substring with only non-repeating characters. *S* = "aaaa" --> 1, as "a" is the longest substring – "aa" or "aaa" or "aaaa" all have the repeating character "a". *S* = "xyz" --> 3, as "xyz" is the longest substring without repeating characters.

5. Longest substring with unique characters of *k* length. Given a string *S* and integer *k* return the string of longest length that contains *k* distinct characters. *S* = "aaa", *k* = 1 --> "a", as "a" is the longest substring that contains *1* distinct character. *S* = "eeefg", *k* = 3 --> "efg", as "efg" is the longest substring that contains *3* distinct characters.

6. Subarrays with *k* different integers. Given an array *L* return the number of subarrays that can be formed with *k* distinct integers. *L* = [1,2,1,2,3], k=2 --> 7, as the # of subarrays with *2* different integers is 7: we can make [1,2], [2,1], [1,2], [2,3], [1,2,1], [2,1,2], [1,2,1,2]. *L* = [1, 2, 1, 3, 4], *k* =3 --> 3, as we can make [1,2,1,3], [2,1,3], [1,3,4], or 3 subarrays with 3 different integers in them.