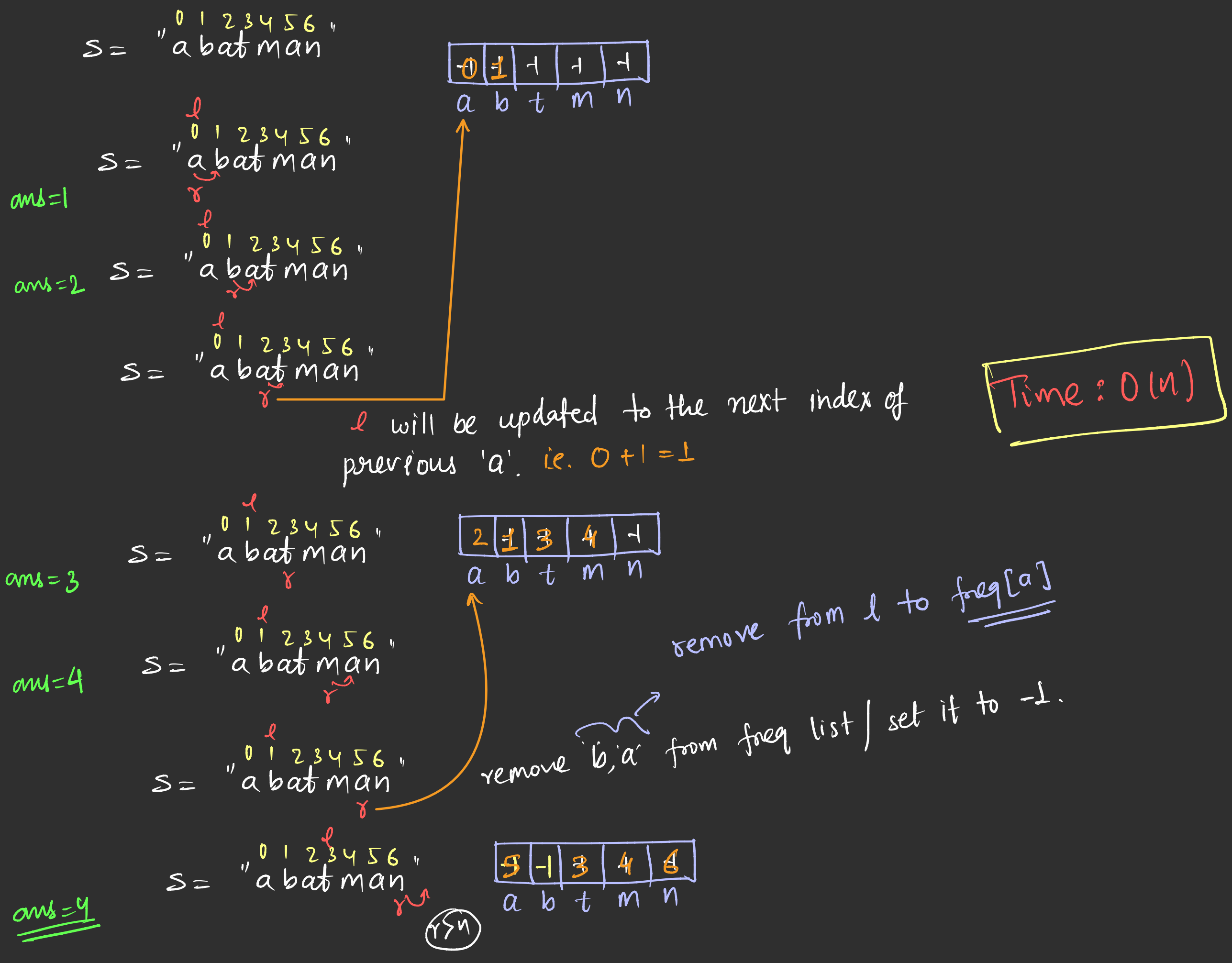
Brute force Approach: → Generate all substring and check for the longest substring without repeating chars.

Time: $O(n^3)$

Optimal solution:

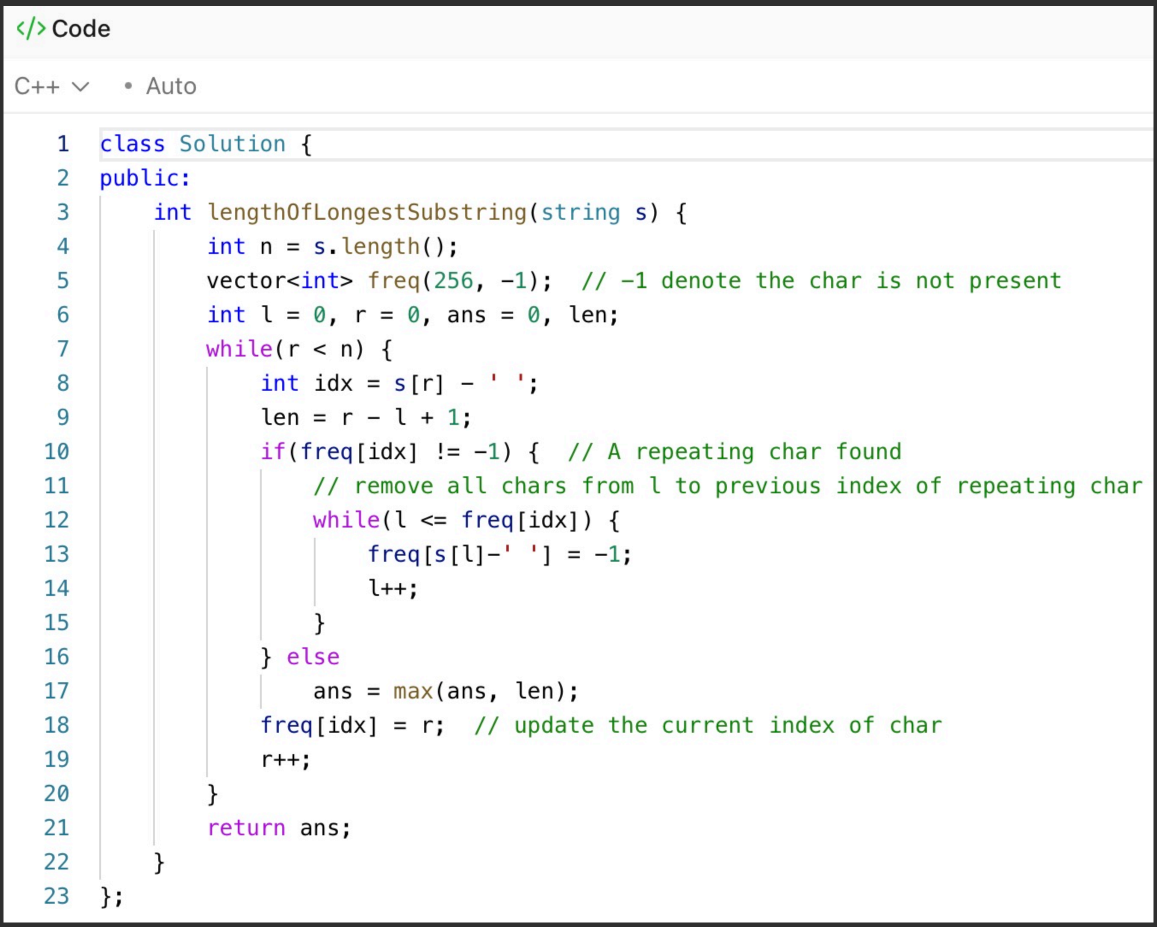


l will be updated to the next index of previous 'a'. ie. 0+1 = 1

Time : O(n)

remove from l to freq[a]

remove 'b,a' from freq list / set it to -1.



```cpp
class Solution {
public:
    int lengthOfLongestSubstring(string s) {
        int n = s.length();
        vector<int> freq(256, -1);  // -1 denote the char is not present
        int l = 0, r = 0, ans = 0, len;
        while(r < n) {
            int idx = s[r] - ' ';
            len = r - l + 1;
            if(freq[idx] != -1) {  // A repeating char found
                // remove all chars from l to previous index of repeating char
                while(l <= freq[idx]) {
                    freq[s[l]-' '] = -1;
                    l++;
                }
            } else
                ans = max(ans, len);
            freq[idx] = r;  // update the current index of char
            r++;
        }
        return ans;
    }
};
```