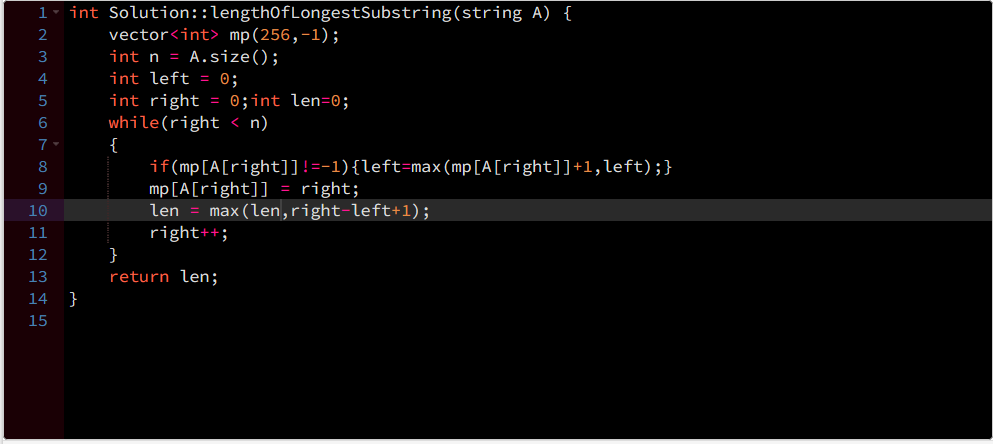
1. Longest substring without repeat



1. Minimum window substring

string minWindow(string s, string t) {

        unordered\_map<char,int> map1;

        unordered\_map<char,int> map2;

        int check=INT\_MAX;

        string result;

        for(char &ch:t)map1[ch]++;

        int slow=0,fast=0,lettercounter=0;

        for(;fast<s.length();fast++)

        {

            char ch=s[fast];

            if(map1.find(ch)!=map1.end())

            {

                map2[ch]++;

            if(map2[ch]<=map1[ch])

                lettercounter++;

            }

            if(lettercounter>=t.length())

            {

                while(map1.find(s[slow])==map1.end()||map2[s[slow]]>map1[s[slow]])

                {

                    map2[s[slow]]--;

                    slow++;

                }

                if(fast-slow+1<check)

                {

                    check=fast-slow+1;

                    result=s.substr(slow,check);

                }

            }

        }

        return result;

    }