1. Repeated DNA Sequences

All DNA is composed of a series of nucleotides abbreviated as A, C, G, and T, for example: “ACGAATTCCG”. When studying DNA, it is sometimes useful to identify repeated sequences within the DNA.

Write a function to find all the 10-letter-long sequences (substrings) that occur more than once in a DNA molecule.

**Example:**

Input: s = "AAAAACCCCCAAAAACCCCCCAAAAAGGGTTT"  
  
Output: ["AAAAACCCCC", "CCCCCAAAAA"]

**解法1** hash + slice window，记录出现的每个子串，找出重复出现的子串

时间复杂度：

空间复杂度：

class Solution {  
public:  
 vector<string> findRepeatedDnaSequences(string s) {  
 vector<string>ans;  
 if(s.size() < 10)return ans;  
 unordered\_map<string, int>mp;  
 for(int i = 0; i < s.size() - 10 + 1; ++i){  
 string tmp = s.substr(i, 10);  
 if(mp[tmp] == 1)ans.push\_back(tmp);  
 mp[tmp] += 1;  
 }  
 return ans;  
 }  
};

**解法2** 将字符串序列看作是数字序列，在本题中将其看做4进制数。对于每一个长度为子串，都可以将其转换为一个数字，以此来优化解法1的空间复杂度

对于字符串：

对于字符串：

class Solution {  
public:  
 vector<string> findRepeatedDnaSequences(string s) {  
 vector<string>ans;  
 if(s.size() < 10)return ans;  
 unordered\_map<long long int, int>mp;  
 unordered\_map<char, int>basis{{'A', 0}, {'G',1},{'T', 2},{'C', 3}};  
 long long int h = 0, d = pow(4, 10);  
 for(int i = 0; i < 10; ++i)h = h \* 4 + basis[s[i]];  
 mp[h] = 1;  
 for(int i = 1; i < s.size() - 10 + 1; ++i){  
 h = (h \* 4 - basis[s[i-1]] \* d) + basis[s[i+10-1]];  
 if(mp[h] == 1)ans.push\_back(s.substr(i, 10));  
 mp[h] += 1;  
 }  
 return ans;  
 }  
};