1. Word Break

Given a **non-empty** string *s* and a dictionary *wordDict* containing a list of **non-empty** words, determine if *s* can be segmented into a space-separated sequence of one or more dictionary words.

**Note:**

* The same word in the dictionary may be reused multiple times in the segmentation.
* You may assume the dictionary does not contain duplicate words.

**Example 1:**

Input: s = "leetcode", wordDict = ["leet", "code"]  
Output: true  
Explanation: Return true because "leetcode" can be segmented as "leet code".

**Example 2:**

Input: s = "applepenapple", wordDict = ["apple", "pen"]  
Output: true  
Explanation: Return true because "applepenapple" can be segmented as "apple pen apple".  
 Note that you are allowed to reuse a dictionary word.

**Example 3:**

Input: s = "catsandog", wordDict = ["cats", "dog", "sand", "and", "cat"]  
Output: false

**解** 用hashset存储词表。用mem数组记录以i为分割点，是否能分割

class Solution {  
public:  
 bool wordBreak(string s, vector<string>& wordDict) {  
 unordered\_map<string, bool>hash;  
 for(string str : wordDict)hash[str] = true;  
 vector<int>mem(s.size(), -1);  
 return check(s, 0, mem, hash);  
 }  
 bool check(const string& s, int start,   
 vector<int>& mem, unordered\_map<string, bool>& hash){  
 if(start >= s.size())return true;  
 if(mem[start] != -1)return mem[start];  
 for(int i = start + 1; i <= s.size(); ++i){  
 if(hash[s.substr(start, i - start)] && check(s, i, mem, hash)){  
 mem[start] = 1;  
 return 1;  
 }  
 }  
 mem[start] = 0;  
 return 0;  
 }  
};