**Word Break**

#include <bits/stdc++.h>

bool wordBreakHelper(const unordered\_set<string>& wordSet, unordered\_map<int, bool>& cache, const string& target, int index) {

if (index >= target.length()) {

return true;

}

if (cache.count(index)) {

return cache[index];

}

for (int j = index; j < target.length(); j++) {

string subs = target.substr(index, j - index + 1);

if (wordSet.count(subs)) {

if (wordBreakHelper(wordSet, cache, target, j + 1)) {

cache[index] = true;

return cache[index];

}

}

}

cache[index] = false;

return cache[index];

}

bool wordBreak(vector<string>& arr, int n, string target) {

unordered\_set<string> wordSet(arr.begin(), arr.end());

unordered\_map<int, bool> cache;

return wordBreakHelper(wordSet, cache, target, 0);

}