17. Letter Combinations of a Phone Number <Medium>

vector<string> letterCombinations(string digits) {

if (digits.empty())

return {};

vector<string> ans;

string path;

dfs(digits, 0, path, ans);

return ans;

}

const vector<string> digitToLetters{"", "", "abc", "def", "ghi",

"jkl", "mno", "pqrs", "tuv", "wxyz"};

void dfs(const string& digits, int i, string& path, vector<string>& ans) {

if (i == digits.length()) {

ans.push\_back(path);

return;

}

for (const char letter : digitToLetters[digits[i] - '0']) {

path.push\_back(letter);

dfs(digits, i + 1, path, ans);

path.pop\_back();

}

}

vector<string> letterCombinations(string digits) {

if (digits.empty())

return {};

vector<string> ans{""};

const vector<string> digitToLetters{"", "", "abc", "def", "ghi",

"jkl", "mno", "pqrs", "tuv", "wxyz"};

for (const char d : digits) {

vector<string> temp;

for (const string& s : ans)

for (const char c : digitToLetters[d - '0'])

temp.push\_back(s + c);

ans = move(temp);

}

return ans;

}