**1763. Longest Nice Substring**

class Solution {

public:

bool isnice(const unordered\_set<char>& hs){

for(char c:hs){

if(islower(c)&& hs.find(toupper(c))==hs.end()){

return false;

}

if(isupper(c)&& hs.find(tolower(c))==hs.end()){

return false;

}

}

return true;

}

string longestNiceSubstring(string s) {

int n=s.length();

if (n<2) return "";

string ans="";

for (int i=0; i<n; i++) {

unordered\_set<char> hs;

for (int j=i; j<n; ++j) {

hs.insert(s[j]);

if (isnice(hs)) {

if (j-i+1 > ans.length()) {

ans = s.substr(i, j - i + 1);

}

}

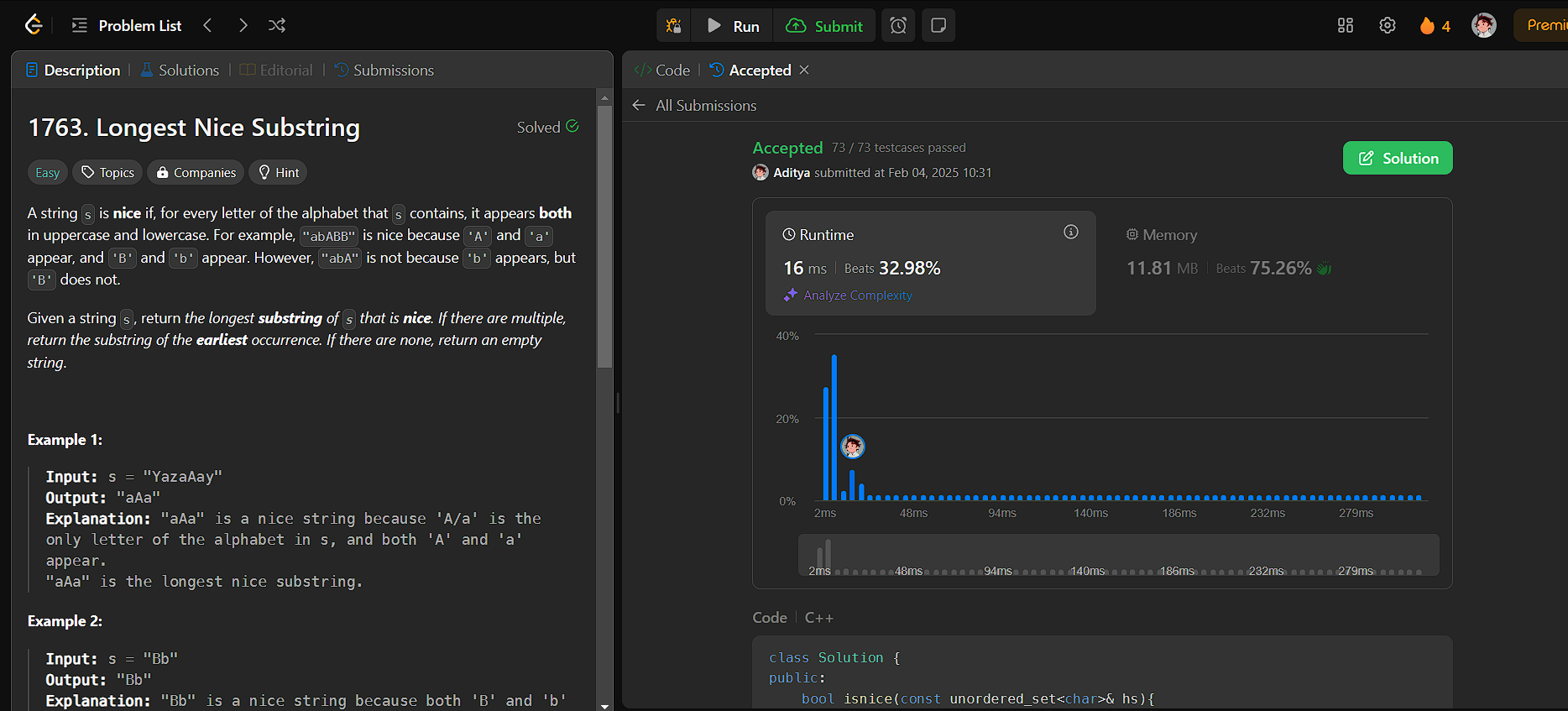
}

}

return ans;

}

};



**190. Reverse Bits**

class Solution {

public: