//Longest Repeated Substring Uva 11512 - GATTACA

//Trie, Find the deepest node which is visited more than once

struct node {

int visited;

bool isEnd;

node \*next[4];

node() {

visited = 0;

isEnd = false;

for(int i = 0; i < 4; i++)

next[i] = NULL;

}};

void build(char str[], int len, node \*current) {

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

int pos = mp[str[i]];

if(current->next[pos] == NULL)

current->next[pos] = new node();

current = current->next[pos];

current→visited++; }

current->isEnd = true;

}

//Longest Repeated Subsequence

//if two LRS count are same, returns the alpha - smaller one

string LRS(node \*current, string past, int \*cnt) {

int pos;

string y;

y += past;

for(int i = 0; i < 4; i++) {

if(current->next[i] != NULL) {

if(current->next[i]->visited > 1) {

string tmp;

tmp += rev[i];

//\*cnt = min(current->next[i]->visited, \*cnt);

int cnt\_next = 1e4;

tmp += LRS(current->next[i], "", &cnt\_next);

if(tmp.size() > y.size()) {

y = tmp;

//\*cnt = min(\*cnt, cnt\_next);

}

else if(tmp.size() == y.size())

if(tmp < y) {

y = tmp;

//\*cnt = min(\*cnt, cnt\_next);

}}}}

return y;

}

//TO DELETE NODES (DEALLOCATE MEMORY), see TRIE delete node function (del())