#include <bits/stdc++.h>

#define mp make\_pair

#define pb push\_back

#define sz(x) (int)x.size()

#define all(x) begin(x), end(x)

#define fi first

#define se second

#define debug(x) cerr << #x << " " << x << '\n'

using namespace std;

using ll = long long;

using pii = pair<int,int>;

using pli = pair<ll,int>;

const int INF = 0x3f3f3f3f, N = 3e5 + 5;

const ll LINF = 1e18 + 5;

constexpr int mod = 1e9 + 7;

char a[N], b[N], rb[N];

char s[N][30], rs[N][30];

int n, val[N];

ll pre[N], suf[N];

struct ACA

{

int nxt[N][26], fail[N], len[N], cnt, end[N];

ll tmp[N][30];

vector <int> to[N];

void clear()

{

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

{

fail[i] = end[i] = len[i] = 0;

to[i].clear();

for(int j=0; j<26; j++)

nxt[i][j] = tmp[i][j+1] = 0;

}

cnt = 0;

}

void insert(char \*s, int v, int n)

{

int p = 0;

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

{

int k = s[i] - 'a';

if(!nxt[p][k]) nxt[p][k] = ++cnt;

p = nxt[p][k];

}

end[p] = v; len[p] = n;

}

void build()

{

queue <int> q;

for(int i=0;i<26;i++) if(nxt[0][i]) q.push(nxt[0][i]);

while(!q.empty())

{

int k = q.front(); q.pop();

for(int i=0;i<26;i++)

{

if(nxt[k][i])

{

fail[nxt[k][i]] = nxt[fail[k]][i];

q.push(nxt[k][i]);

}

else nxt[k][i] = nxt[fail[k]][i];

}

}

}

void go(char \*s, ll \*v, int n)

{

int p = 0;

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

{

p = nxt[p][s[i]-'a'];

for(int j=p; j; j=fail[j])

v[i] += end[j];

}

}

void dfs(int u)

{

for(int i=len[u]; i>0; i--) tmp[u][i] += end[u];

for(int v : to[u])

{

for(int i=26; i>0; i--) tmp[v][i] += tmp[u][i];

dfs(v);

}

}

}ac, ac2;

void solve()

{

scanf("%s%s", a, b);

scanf("%d", &n);

int x = strlen(a), y = strlen(b);

for(int i=0; i<y; i++) rb[i] = b[y-i-1];

ac.clear(); ac2.clear();

for(int i=1; i<=n; i++)

{

scanf("%s%d", s[i], val+i);

int t = strlen(s[i]);

ac.insert(s[i], val[i], t);

for(int j=0; j<t; j++) rs[i][j] = s[i][t-j-1];

ac2.insert(rs[i], val[i], t);

}

ac.build(); ac2.build();

for(int i=0; i<x; i++) pre[i] = 0;

for(int i=0; i<y; i++) suf[i] = 0;

ac.go(a, pre, x); ac2.go(rb, suf, y);

ll ans = 0;

int p = 0;

for(int i=1; i<x; i++) pre[i] += pre[i-1];

for(int i=1; i<y; i++) suf[i] += suf[i-1];

for(int i=1; i<=ac.cnt; i++) ac.to[ac.fail[i]].pb(i);

ac.dfs(0);

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

{

p = ac.nxt[p][a[i]-'a'];

for(int j=0; j<y; j++)

{

ll cur = pre[i] + suf[y-j-1];

int pp = p;

for(int k=0; k<25&&j+k<y; k++)

{

pp = ac.nxt[pp][b[j+k]-'a'];

cur += ac.tmp[pp][k+2];

}

if(cur>ans) ans = cur;

}

}

printf("%lld\n", ans);

}

int main()

{

int T; scanf("%d", &T);

while(T--) solve();

return 0;

}