#include <cstdio>

#include <vector>

#include <queue>

#include <algorithm>

#include <bits/stdc++.h>

using namespace std;

vector <pair<int, int> > v[2005];

int dist[16][2005];

int dp[1<<16][16];

int ub[16];

bool in\_q[2005];

int pow2[16];

queue <int> q;

const int INF = 1<<26;

int main(){

freopen("ubuntzei.in", "r", stdin);

freopen("ubuntzei.out", "w", stdout);

int i,j,k,l,p,x,y,z,m,n;

scanf("%d %d", &n, &m);

scanf("%d", &k);

for(i = 1;i <= k;i++){

scanf("%d", &ub[i]);

}

for(i = 1;i <= m;i++){

scanf("%d %d %d", &x, &y, &z);

v[x].push\_back({y, z});

v[y].push\_back({x, z});

}

ub[0] = 1;

for(i = 0;i <= k;i++){

for(j = 1;j <= n;j++){

dist[i][j] = INF;

}

dist[i][ub[i]] = 0;

q.push(ub[i]);

in\_q[ub[i]] = true;

while(q.empty() == false){

x = q.front();

in\_q[x] = false;

q.pop();

for(auto it : v[x]){

if(dist[i][it.first] > dist[i][x] + it.second){

dist[i][it.first] = dist[i][x] + it.second;

if(in\_q[it.first] == 0){

in\_q[it.first] = true;

q.push(it.first);

}

}

}

}

}

pow2[0] = 1;

for(i = 1;i <= 15;i++){

pow2[i] = pow2[i-1]<<1;

}

int limit = pow2[k];

for(i = 0;i < limit;i++){

for(j = 0;j <= k;j++){

dp[i][j] = INF;

}

}

dp[0][0] = 0;

for(i = 1;i < limit;i++){

for(j = 0;j < k;j++){

if(i&pow2[j]){

for(l = 0;l <= k;l++){

dp[i][j + 1] = min(dp[i][j + 1], dp[i-pow2[j]][l] + dist[l][ub[j+1]]);

}

} } }

int ans = INF;

for(i = 1;i <= k;i++){

ans = min(ans, dp[limit-1][i] + dist[i][n]);

}

if(ans == INF){ ans = dist[0][n]; }

printf("%d", ans);

return 0;

}