#include <cstdio>

#include <algorithm>

using namespace std;

#define inf 1000000000

struct node

{

int d, p;

} h[50010];

bool operator < (node a, node b)

{

return a.d > b.d;

}

int d[20010];

int e[50010][3];

int \*g[20010], \*w[20010], deg[20010];

int main()

{

int T, l, n, m, s, t, p, i;

scanf("%d", &T);

for (l = 1; l <= T; l++)

{

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

memset(deg, 0, sizeof(deg));

for (i = 0; i < m; i++)

{

scanf("%d%d%d", e[i], e[i]+1, e[i]+2);

deg[e[i][0]]++;

deg[e[i][1]]++;

}

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

{

g[i] = new int[deg[i]+1];

w[i] = new int[deg[i]+1];

deg[i] = 0;

}

for (i = 0; i < m; i++)

{

g[e[i][0]][deg[e[i][0]]] = e[i][1];

g[e[i][1]][deg[e[i][1]]] = e[i][0];

w[e[i][0]][deg[e[i][0]]++] = e[i][2];

w[e[i][1]][deg[e[i][1]]++] = e[i][2];

}

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

d[i] = inf;

m = 0;

d[s] = 0;

h[m].p = s;

h[m++].d = 0;

while (m)

{

p = h[0].p;

if (p == t)

break;

pop\_heap(h, h+m--);

for (i = 0; i < deg[p]; i++)

if (d[g[p][i]] > d[p]+w[p][i])

{

h[m].p = g[p][i];

h[m++].d = d[g[p][i]] = d[p]+w[p][i];

push\_heap(h, h+m);

}

}

if (d[t] < inf)

printf("Case #%d: %d\n", l, d[t]);

else

printf("Case #%d: unreachable\n", l);

}

return 0;

}