int get\_map\_list\_index(adj\_list\_t list[], traffic\_sig\_jn \*ptr)

{

int i=0;

for(i=0;i<list->num\_map\_extn;i++)

{

if(ptr == list[i]->src\_tsj )

{

break;

}

}

return i;

}

void get\_direction(map\_extn\_t \*source, map\_extn\_t \*dest, route\_t \*directions, graph\_tsj\_t \*g)

{

traffic\_sig\_jn\_t \*target, \*start;

int cost[g->num\_tsj][g->num\_tsj], i, j, m, weight=0, flag[g->num\_tsj], min, count, prev[g->num\_tsj], diatance[g->num\_tsj], d;

traffic\_sig\_jn\_t \*temp;

direction->steps = (traffic\_sig\_jn\_t \*)traffic\_malloc(sizeof(traffic\_sig\_jn\_t)\*g->num\_tsj);

for(i=0;i<g->num\_tsj;i++) // initializing cost matrix

{

for(j=0;j<g->num\_tsj;j++)

{

cost[i][j] = 99;

flag[i]=0;

distance[i]=99;

prev[i]=-1;

}

}

for(i=0;i<g->num\_tsj;i++) // creating cost matrix

{

temp = g->tsj\_adj\_list[i]->head;

for(j=0;j<4;j++)\

{

k=get\_map\_list\_index(g->tsj\_adj\_list, temp->dest\_jn\_p);

cost[i][k] = temp->street\_p->lanes[0][0]->max\_vehicles;

temp = temp->next;

}

}

target = dest->adj\_signal;

start = source->adj\_signal;

weight = cost[get\_map\_list\_index(g->tsj\_adj\_list, start)][get\_map\_list\_index(g->tsj\_adj\_list, target)];

flag[get\_map\_list\_index(g->tsj\_adj\_list, start]=1;

distance[get\_map\_list\_index(g->tsj\_adj\_list, start]=0;

while(flag[get\_map\_list\_index(g->tsj\_adj\_list, target]!=0)

{

min = 99;

m=0;

for(i=1;i<g->num\_tsj;i++)

{

d= distance[get\_map\_list\_index(g->tsj\_adj\_list, start]+cost[get\_map\_list\_index(g->tsj\_adj\_list, start][i];

if(d<distance[i]&&flag[i]==0)

{

distance[i] = d;

prev[i] = get\_map\_list\_index(g->tsj\_adj\_list, start];

}

if(min>distance[i] && flag[i]==0)

{

min = distance[i];

m = i;

}

}

start = g->tsj\_adj\_list[m];

flag[get\_map\_list\_index(g->tsj\_adj\_list, start]=1;

}

start = target;

j=0;

while(get\_map\_list\_index(g->tsj\_adj\_list, start]!=-1))

{

direction->steps [j++] = start;

start = g->tsj\_adj\_list[prev[get\_map\_list\_index(g->tsj\_adj\_list, start]];

}

direction->steps [j]=NULL;

strrev(direction->steps );

directions->no\_of\_steps++;

}