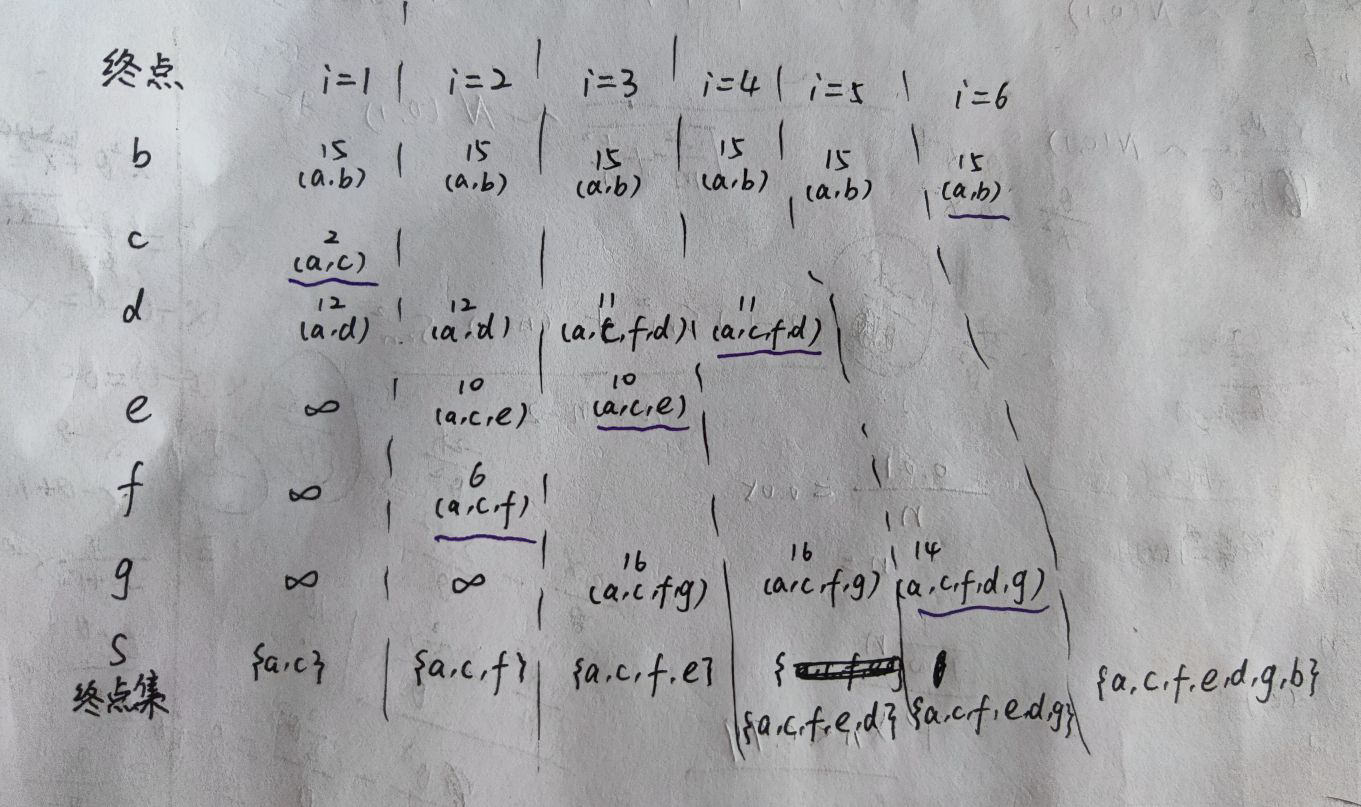
应用题



算法题

(3)

void shortestmax(AMGraph G, int v0)

{

n = G.vexnum;

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

{

S[v] = false;

D[v] = G.arcs[v0][v];

if (D[v] < MaxInt)

Path[v] = v0;

else Path[v] = -1;

}

S[v0] = true;

D[v0] = 0;

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

{

min = MaxInt;

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

if (!S[w] && D[w] < min)

{

v = w;

min = D[w];

}

S[v] = true;

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

if (!S[w] && (D[v] + G.arcs[v][w] < D[w]))

{

D[w] = D[v] + G.arcs[v][w];

Path[w] = v;

}

}

max = D[0];

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

if (D[i] > max)

{

max = D[i];

k = i;

}

return k;

}

(5)

int visited[MAXSIZE]=false;

int length(ALGragh G, int i, int j, int k)

{

if (i == j && k = 0)

return 1;

else if (k > 0)

{

visited[i] = true;

for (p = G.vertices[i].firstarc;p;p = p->nextarc)

{

v = p->adjvex;

if (!visited[v])

if (length(G, v, j, k - 1))

return 1;

}

visited[i] = false;

}

return 0;

}