

```
// 「Cで学ぶデータ構造とアルゴリズム」(西原清一) オーム社, 2008
// 図5・26 (p.126) Dijkstra アルゴリズム
// 改訂版
```

```
#include<stdio.h>
#define      N      4
#define      M      999 /*無限大*/

int w[N][N]={ {0, 9, 4, M}, {M, 0, M, 9}, {M, M, 0, 2}, {3, 1, M, 0} };
int d[N]={0, 0, 0, 0}, S[N]={0, 0, 0, 0};

void add(int u, int *S) {S[u] = 1;}

int remain()          // V-S is still not empty?
{   int i;
    for (i=0; i<N; i++) if (S[i] == 0) return 1;
    return 0;
}

int select_min()      // select u such that (min d[u] && i∈V-S)
{   int min=M, i, u;
    for (i=0; i<N; i++) {
        if (S[i] == 0 && d[i] < min) {
            min = d[i]; u = i;
        }
    }
    return u;
}

int member(int u, int x)      // <u, i>∈E && i∈V-S
{   if (w[u][x] != M && S[x] == 0)
        return 1;
    else
        return 0;
}

void Dijkstra(int p)
{   int i, u, x, k;
    add(p, S);
    for (i=0; i<N; i++) d[i] = w[p][i];
```

```

while (remain()) {
    u = select_min();
    add(u, S);
    for (x=0; x<N; x++) {
        if (member(u, x)) {
            if ((k = d[u]+w[u][x]) < d[x]) d[x] = k;}
        }
    }
}

main()
{
    int i, j=0;
    Dijkstra(j);
    printf("shortest distance from node %d\n", j);
    for (i=0; i<N; i++)
        printf(" to node %d: %d\n", i, d[i]);
}

```