



به نام خدا
شبکه های کامپیوتری



تمرین کامپیوتری شماره ۳

(الگوریتم های مسیریابی در شبکه)

اعضای گروه :

۸۱۰۱۹۸۴۱۰

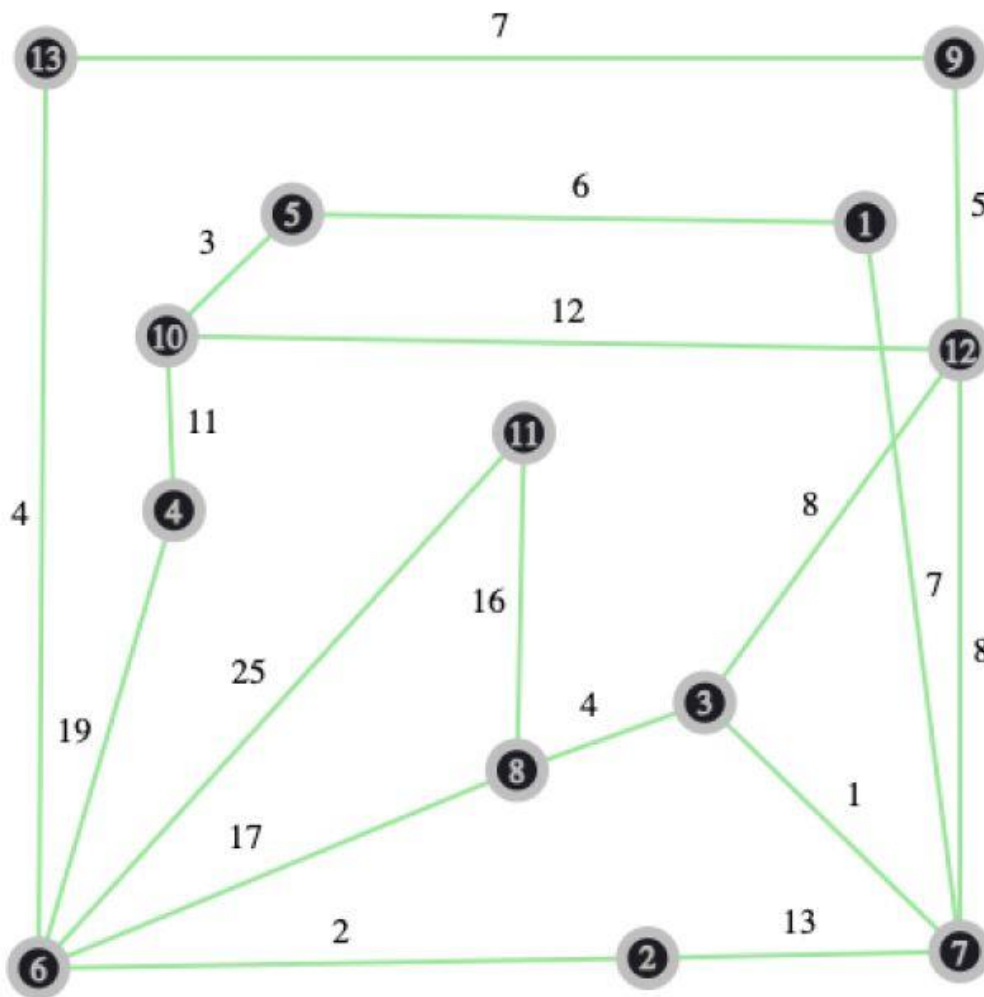
محمد سعادت

۸۱۰۱۹۸۴۳۶

محمد عراقی

مقدمه

در این پروژه، به پیاده سازی الگوریتم های distance vector و link state پرداخته شده و برای یک شبکه کوچک، این دو الگوریتم اجرا شده و نتایج آن ها از لحاظ زمانی بررسی شده است .



الگوریتم ها

Distance Vector

```
void handle_dvrp_command(vector<string> nodes)
{
    vector<vector<int>> next_hop(number_of_nodes + 1, vector<int>(number_of_nodes + 1, NO_EDGE));
    vector<vector<string>> shortest_path(number_of_nodes + 1, vector<string>(number_of_nodes + 1, ""));

    auto start = chrono::high_resolution_clock::now();

    for (int i = 0; i < number_of_nodes + 1; i++)
        next_hop[i][i] = i;

    for (int i = 1; i < number_of_nodes + 1; i++)
        for (int j = 1; j < number_of_nodes + 1; j++)
            if (i == j)
                shortest_path[i][i] = to_string(i);
            else if (topology[i][j] != NO_EDGE)
                shortest_path[i][j] = to_string(i) + NUMBER_SEPARATOR + to_string(j) + NUMBER_SEPARATOR;

    for (int i = 1; i < number_of_nodes + 1; i++)
        for (int j = 1; j < number_of_nodes + 1; j++)
            if (topology[i][j] != NO_EDGE)
                next_hop[i][j] = i;

    vector<vector<int>> distance = topology;
    for (int k = 1; k < number_of_nodes + 1; k++)
        for (int i = 1; i < number_of_nodes + 1; i++)
            for (int j = 1; j < number_of_nodes + 1; j++)
                if (distance[i][k] != NO_EDGE and distance[k][j] != NO_EDGE and distance[i][k] + distance[k][j] < distance[i][j])
                {
                    distance[i][j] = distance[i][k] + distance[k][j];
                    shortest_path[i][j] = shortest_path[i][k] + NUMBER_SEPARATOR + shortest_path[k][j] + NUMBER_SEPARATOR;
                    next_hop[i][j] = next_hop[i][k] != i ? next_hop[i][k] : k;
                }

    auto end = chrono::high_resolution_clock::now();
    time_taken = time_taken + chrono::duration_cast<chrono::nanoseconds>(end - start).count();
    time_taken *= 1e-9;
}
```

Link State

```

pair<vector<int>, vector<int>> dijkstra(int src)
{
    vector<int> parent(number_of_nodes , src);
    vector<int> dist(number_of_nodes , NO_EDGE);
    vector<bool> flag(number_of_nodes , false);

    auto start = chrono::high_resolution_clock::now();

    parent[src] = -1;
    dist[src] = 0;

    for (int c = 0; c < number_of_nodes - 1; c++)
    {
        int min_val = NO_EDGE, min_index;
        for (int i = 0; i < number_of_nodes ; i++)
            if (flag[i] == false and dist[i] < min_val)
            {
                min_val = dist[i];
                min_index = i;
            }
        flag[min_index] = true;
        for (int v = 0; v < number_of_nodes ; v++)
            if (!flag[v] and topology[min_index + 1][v + 1] != NO_EDGE and dist[min_index] + topology[min_index + 1][v + 1] < dist[v])
            {
                parent[v] = min_index;
                dist[v] = dist[min_index] + topology[min_index + 1][v + 1];
            }
        show_lsrp_iteration_table(dist, c + 1);
    }

    auto end = chrono::high_resolution_clock::now();
    time_taken = time_taken + chrono::duration_cast<chrono::nanoseconds>(end - start).count();
    time_taken *= 1e-9;
    return {parent, dist};
}
    
```

نتایج

برای این شبکه، میتوان مشاهده کرد که با اجرای دستور dvrp و الگوریتم distance vector، حدود

0.00072s زمان قبل از تغییر و 0.000070s پس از تغییر نیاز داریم.

```

Routing table for node 13:
Dest      Next Hop      Dist      Shortest path
-----
1         6              26        [13->6->2->7->1]
2         6              6          [13->6->2]
3         6              20         [13->6->2->7->3]
4         6              23         [13->6->4]
5         9              27         [13->9->12->10->5]
6         6              4          [13->6]
7         6              19         [13->6->2->7]
8         6              21         [13->6->8]
9         9              7          [13->9]
10        9              24         [13->9->12->10]
11        6              29         [13->6->11]
12        9              12         [13->9->12]
13        13             0          [13]

dvrp time is : 0.000072597 sec

```

```

Routing table for node 13:
Dest      Next Hop      Dist      Shortest path
-----
1         6              26        [13->6->2->7->1]
2         6              6          [13->6->2]
3         6              20         [13->6->2->7->3]
4         6              23         [13->6->4]
5         9              27         [13->9->12->10->5]
6         6              4          [13->6]
7         6              19         [13->6->2->7]
8         6              21         [13->6->8]
9         9              7          [13->9]
10        9              24         [13->9->12->10]
11        6              29         [13->6->11]
12        9              12         [13->9->12]
13        13             0          [13]

dvrp time is : 0.000070619 sec

```

اما برای همین شبکه، پس از اجرای دستور lsrp و الگوریتم link state، به 0.00016s زمان قبل از تغییر و به 0.00013s زمان پس از تغییر توپولوژی نیاز داریم.

```

Routing table for node 13:
Path: [s]->[d]      Min_Cost      Shortest Path
-----
[13]->[1]           26           13->6->2->7->1
[13]->[2]            6           13->6->2
[13]->[3]           20           13->9->12->3
[13]->[4]           23           13->6->4
[13]->[5]           27           13->9->12->10->5
[13]->[6]            4           13->6
[13]->[7]           19           13->6->2->7
[13]->[8]           21           13->6->8
[13]->[9]            7           13->9
[13]->[10]          24           13->9->12->10
[13]->[11]          29           13->6->11
[13]->[12]          12           13->9->12

lsrp time is : 0.000132 sec

```

```

Routing table for node 13:
Path: [s]->[d]      Min_Cost      Shortest Path
-----
[13]->[1]           26           13->6->2->7->1
[13]->[2]            6           13->6->2
[13]->[3]           20           13->9->12->3
[13]->[4]           23           13->6->4
[13]->[5]           27           13->9->12->10->5
[13]->[6]            4           13->6
[13]->[7]           19           13->6->2->7
[13]->[8]           21           13->6->8
[13]->[9]            7           13->9
[13]->[10]          24           13->9->12->10
[13]->[11]          29           13->6->11
[13]->[12]          12           13->9->12

lsrp time is : 0.000164605 sec

```

در نتیجه، میتوان گفت که الگوریتم link state برای شبکه ما با مقدار نسبتاً خوبی بهتر کار میکند، اما با تغییر و بزرگتر شدن شبکه میتوان انتظار داشت در مواردی distance vector بهتر و سریعتر کار کند.