





更新D的邻接点下

---

计算D的邻接点Y的新dist值

---

temp =



temp <= dist[i]

*rwed*



```
1 void Dijkstra(int start)
2 {
3     init();
4     dist[start] = 0;
5     while(true){
6         int min = INT_MAX;
7         int min_index = -1;
8         for( int i = 1; i <= n; i++){
9             if(min > dist[i] && !collected[i]){
10                 min = dist[i];
11                 min_index = i;
12             }
13         } // 14-19行为找出最近的未被收录的点
14         if(min_index == -1){ // 如果找不到, 跳出循环
15             break;
16         }
17         collected[min_index] = true; // 将该点收录,
18         for( int i = 1; i <= n; i++){ // 遍历该点的邻接点
19             if(map[min_index][i] != -1 && !collected[i]){
20                 int temp = dist[min_index] + map[min_index][i];
21                 if(temp < dist[i]){
22                     dist[i] = temp;
23                     path[i] = min_index;
24                 }
25             }
26         }
27     }
28 }
```









d

i

s

t

path

codeledated













1

4











for use

faise

*for use*





























A large, solid green circle occupies the entire frame. In the center of this circle is a bold, black, sans-serif capital letter 'D'. The 'D' is perfectly centered both horizontally and vertically within the green circle.

**D**

A large yellow circle with a black letter 'Y' in the center.

**Y**

A large, solid red circle is centered on a white background. Inside the circle, a bold, black capital letter 'T' is centered. The 'T' has a thick vertical stem and a horizontal crossbar.

**T**

A large, solid yellow circle serves as the background for the entire image.

**Q**



P



10



























**Y**

A large yellow circle with a black outline, containing a bold black letter 'T' in the center.

**T**