





```
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 }
```



























A large yellow circle with a black outline, containing a bold black letter 'D' in the center.

**D**



A large, solid yellow circle is centered on a white background. Inside the circle, a bold, black, sans-serif capital letter 'Y' is centered. The 'Y' is composed of three thick strokes: two diagonal strokes meeting at a point in the upper half, and a single vertical stroke extending from that point to the bottom half of the circle.

**Y**

A large yellow circle with a black letter 'T' in the center.

**T**

A large, solid yellow circle occupies the entire frame. In the center of this circle is a bold, black, sans-serif capital letter 'Q'. The 'Q' is slightly offset to the right, with its tail extending towards the bottom right edge of the circle.

**Q**



P



10















d

i

s

t

path

codeledated













IN F

1234













*for use*

*for use*

faise

*asce*

↑

↑

↑

↑



计算P的邻接点D的新dist值

---

temp =



temp < dist[i] =

*rwed*

NEW

IVE

将点P收录到集合中

---







**D**