



















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, the letter 'Y' is written in a bold, black, sans-serif font. The 'Y' is positioned in the center of the circle, with its top arms extending towards the upper left and right edges, and its stem pointing downwards towards the bottom edge.

Y

A large yellow circle with a black outline, containing a bold 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 perfectly centered both horizontally and vertically.

Q

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

P



10















d

i

s

t

path

codeledated

















1234













for use

for use

for use

for use

初始化起始点



```
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 }
```


for use

IVE

$\min = 0$

$\min_index = P$

找到与起始点最近且未被收录的点 P



P

for use