SPFA算法代码实现

主讲人：邓哲也

SPFA算法实现

- SPFA算法在实现时，需要用到以下四个数组：
- （1）dist[n] 数组存着当前源点到每个点的最短路。
- （2）path[n] 数组的含义同 dijkstra 算法中的 path 数组。
- （3）Q[] 队列数组存放待更新的节点。
- （4）inQ[n] 数组表示当前这个点是否在队列里。

SPFA算法代码实现

- 为了获得更好的时间复杂度，我们选择邻接表来存图。

```
struct edge {
    int v, w, next;
}e[M];
int h[N], ee, dist[N], m, n, inq[N], q[QSIZE];
void addedge(int u, int v, int w) {
    e[ee] = (edge){v, w, h[u]};
    h[u] = ee ++;
}
```

SPFA算法代码实现

- 主体部分：
- ```
  void spfa(int u0) {
      memset(dist, 63, sizeof(dist));
      dist[u0] = 0;
      inq[u0] = 1;
      int head = 0, tail = 1;
      q[0] = u0;
  ```
  ```

SPFA算法代码实现

- 主体部分：

```
while (head != tail) {
    int u = q[head ++];
    head %= QSIZE;
    inq[u] = 0;
    for (int i = h[u]; i != -1; i = e[i].next) {
        int v = e[i].v;
        if (dist[v] > dist[u] + e[i].w) {
            dist[v] = dist[u] + e[i].w;
            if (!inq[v]) {
                q[tail ++] = v;
                tail %= QSIZE;
                inq[v] = 1;
            }
        }
    }
}
}
```

下节课再见