

Dijkstra算法堆 优化代码实现



主讲人：邓哲也



P0J2387 Til the Cows Come Home

- 给出一个 n 个点， m 条无向边（带非负边权）的图。
- 求出从 1 走到 n 的最短路。
- $n \leq 1000$, $m \leq 2000$, 有重边。

- 样例输入:

- 5 5
- 1 2 20
- 2 3 30
- 3 4 20
- 4 5 20
- 1 5 100

- 样例输出:

- 90

Dijkstra算法堆优化代码实现

- 因为堆优化的时间复杂度是 $O((n + m) \log n)$ ，所以这时我们就不该用邻接矩阵存图，不然空间复杂度变成了 $O(n^2)$ ，优化时间复杂度没意义了。
- ```
struct edge {
```
- ```
    int v, w, next;
```
- ```
} e[M];
```
- ```
int h[N], ee;
```

Dijkstra算法堆优化代码实现

- 加边操作:
- `void addedge(int u, int v, int w) {`
- `e[ee] = (edge){v, w, h[u]};`
- `h[u] = ee ++;`
- `}`

Dijkstra算法堆优化代码实现

- 在之前堆的章节中，我们讲过了用 STL 中的 `priority_queue` 来实现堆。这次，我们介绍另一个工具，叫做 `set`，它也可以起到堆的作用。
- `set` 可以看作一个从小到大有序的序列，支持插入，删除，查找，且复杂度均为 $O(\log n)$ 。
- 这里我们只需要使用插入，查询头元素和删除头元素。

Dijkstra算法堆优化代码实现

主体部分:

```
# include <set>
```

```
set < pair<int, int> > heap;
```

```
void dijkstra(int u0) {
```

```
    memset(dist, 63, sizeof(dist));
```

```
    dist[u0] = 0;
```

```
    for (int i = 1; i <= n; i ++)
```

```
        heap.insert(make_pair(dist[i], i));
```

Dijkstra算法堆优化代码实现

```
while (heap.size() > 0) {  
    int d = heap.begin()->first;  
    int u = heap.begin()->second;  
    heap.erase(heap.begin());  
    if (d > dist[u]) continue;  
    for (int i = h[u]; i != -1; i = e[i].next) {  
        if (dist[u] + e[i].w < dist[e[i].v]) {  
            dist[e[i].v] = dist[u] + e[i].w;  
            heap.insert(make_pair(dist[e[i].v], e[i].v));  
        }  
    }  
}
```

Di jkstra算法堆优化代码实现

- 快去自己动手试一试吧！
- 也可以尝试一下用 `priority_queue` 实现一下哦！

下节课再见