Prim算法堆优化代码实现

主讲人：邓哲也

Prim算法堆优化主要代码

这里我们使用邻接表来存图。

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

Prim算法堆优化主要代码

用 STL 中的 set 来代替堆使用。

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

Prim算法堆优化主要代码

代码主体：

```
int prim(int u0) {
    for (int i = 1;i <= n;i ++) low[i] = 0x3f3f3f3f,
S[i] = 0;
    low[u0] = 0;
    for (int i = 1;i <= n;i ++)
        heap.insert(make_pair(low[i], i));
```

Prim算法堆优化主要代码

代码主体：

```
while (heap.size() > 0) {
    int d = heap.begin()->first;
    int u = heap.begin()->second;
    heap.erase(heap.begin());
    if (d > low[u]) continue;
    S[u] = 1;
```

Prim算法堆优化主要代码

代码主体：

```
    for (int i = h[u]; i != -1; i = e[i].next) {
        if (!S[e[i].v] && e[i].w < low[e[i].v]) {
            low[e[i].v] = e[i].w;
            heap.insert(make_pair(low[e[i].v],
e[i].v));
        }
    }
}
```

Prim算法堆优化主要代码

代码主体：

```
    int ans = 0;
    for (int i = 1; i <= n;i ++)
        ans += low[i];
    return ans;
}
```

试一试吧！—— HDU 1233

输入格式

每个测试用例的第1行给出村庄数目N（＜ 100 ）；随后的N(N-1)/2行对应村庄间的距离，每行给出一对正整数，分别是两个村庄的编号，以及此两村庄间的距离。为简单起见，村庄从1到N编号。

输出格式

对每个测试用例，在1行里输出最小的公路总长度。

下节课再见