**[pku3268 dij+heap](http://www.cppblog.com/qywyh/articles/28653.html)**

写了个比较通用的堆，可直接用作优先队列

#include <iostream>  
http://www.cppblog.com/Images/OutliningIndicators/None.gifusing namespace std;  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifconst int INF = 1 << 28;  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint adj[1001][1001], adjw[1001][1001], na[1001];  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint n, m, x;  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/None.gif//heap sink,swim,getmin,insert参数均为外部编号,wt为其权值  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint heap[100001], id[100001], hsize;  
http://www.cppblog.com/Images/OutliningIndicators/None.gifint \*key;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifvoid init(int s, int \*wt) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    hsize = s;   
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    key = wt;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    for (i=1; i<=hsize; i++) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        heap[i] = i;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        id[i] = i;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifvoid swim(int u) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int p = id[u], q = p >> 1, ku = key[u];  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    while (q && ku < key[heap[q]]) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        id[heap[q]] = p;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        heap[p] = heap[q];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        p = q;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        q = p >> 1;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    id[u] = p;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    heap[p] = u;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifvoid sink(int u) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int p = id[u],q = p << 1, ku = key[u];  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    while (q <= hsize) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if (q < hsize && key[heap[q+1]] < key[heap[q]]) q++;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if (key[heap[q]] >= ku) break;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        id[heap[q]] = p;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        heap[p] = heap[q];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        p = q;   
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        q = p << 1;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    id[u] = p;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    heap[p] = u;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifint getmin() http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int ret = heap[1];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    id[ret] = -1;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    id[heap[hsize]] = 1;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    heap[1] = heap[hsize];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    hsize--;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    sink(heap[1]);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return ret;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifvoid insert(int u) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    heap[++hsize] = u;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    id[u] = hsize;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    swim(u);  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifvoid build() http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    for (i=hsize/2; i>0; i--) sink(heap[i]);  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifbool isEmpty() http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return hsize == 0;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifint dijkstraHeap(int beg, int end=-1) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i, j, k, u, v, w;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int dist[1001], chk[1001];  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    for (i=1; i<=n; i++) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        dist[i] = INF;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        chk[i] = 0;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    init(n, dist);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    dist[beg] = 0; swim(beg);  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    while (!isEmpty()) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        u = getmin();  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if (u == end) break;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        chk[u] = 1;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif        for (i=0; i<na[u]; i++) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            v = adj[u][i];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif            w = adjw[u][i];  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif            if (dist[v] > dist[u] + w) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                dist[v] = dist[u] + w;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif                swim(v);  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif            }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif        }  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    if (end == -1) return dist[n];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return dist[end];  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}  
http://www.cppblog.com/Images/OutliningIndicators/None.gif  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedBlock.gifint main() http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int i, j, k, u, v, w;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int val[1001];  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    scanf("%d%d%d", &n, &m, &x);  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    for (i=0; i<m; i++) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        scanf("%d%d%d", &u, &v, &w);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        adj[u][na[u]] = v;   
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        adjw[u][na[u]] = w;  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        na[u]++;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif     
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    dijkstraHeap(x);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    memcpy(val, key, sizeof(val));  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif      
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    int ans = 0;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockStart.gifhttp://www.cppblog.com/Images/OutliningIndicators/ContractedSubBlock.gif    for (i=1; i<=n; i++) http://www.cppblog.com/Images/dot.gif{  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        int tmp = dijkstraHeap(i,x);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif        if (tmp+val[i] > ans) ans = tmp + val[i];  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif    }  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif      
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    printf("%d\n", ans);  
http://www.cppblog.com/Images/OutliningIndicators/InBlock.gif    return 0;  
http://www.cppblog.com/Images/OutliningIndicators/ExpandedBlockEnd.gif}