```
from heapclass import heap
#prim using heap! yay!

def load_graph(filename):
    g = {}
    num_nodes = None
    with open(filename, 'r') as fin:
        num_nodes = int(fin.readline().split()[0])
        for line in fin:
            (v1,v2,c) = [int(x) for x in line.split()]
            if not v1 in g:
                g[v1] = []
            if not v2 in g:
                g[v2] = []
            g[v1].append((v2,c))
            g[v2].append((v1,c))
    return g, num_nodes


#g = {n:(v, cost), (v2, cost2)}
def prim(g, num_nodes):
    s = 1
    explored = [False] * (num_nodes+1)
    h = heap()
    #init heap
    cost = [float('inf')]*(num_nodes+1)
    cost[s] = 0
    for i in range(1, num_nodes+1):
        h.insert((cost[i], i))

    cost_sum = 0

    while(h.size() > 0):
        (c, n) = h.get_min()
        explored[n] = True
        cost_sum += cost[n]
        print 'sum inc by',cost[n]
        for (v, c) in g[n]:
            if not explored[v] and c < cost[v]:
                #update
                h.delete_item((cost[v], v))
                cost[v] = c
                h.insert((c, v))

    return cost_sum

def mst_cost(filename):
    g,num_nodes = load_graph(filename)
    s = prim(g, num_nodes)
    print 'sum is',s

mst_cost('edges.txt')
```