```
from heapq import heappop, heappush

def prim(G, s):
    P, Q = {}, [(0, None, s)]
    while Q:
        _, p, u = heappop(Q)
        if u in P: continue
        P[u] = p
        print G[u].items()
        for v, w in G[u].items():
            heappush(Q, (w, u, v))
    return P

def test_prim():
    G = {
      0: {1:1, 2:3, 3:4},
      1: {0:1, 2:5},
      2: {0:3, 1:5, 3:2},
      3: {2:2, 0:4}
    }
    print prim(G, 0)

    G = {
      'a': {'b':1, 'f':2},
      'b': {'a':1, 'c':1},
      'c': {'b':1},
      'd': {'f':1, 'e':2},
      'e': {'d':2, 'g':1},
      'f': {'a':2, 'd':1},
      'g': {'e':1}
    }
    print prim(G, 'a')

test_prim()
```

Sedgewick, R. *Algorithms*, sf. 409

Heatland, Python Algorithms