```
Dijkstra_ShortestPath ( Graph G, Node source ) {
    for each vertex v {
            d[v] = infinity;
            pi[v] = null pointer;
            S[v] = 0;
     }
    d[source] = 0;
    put all vertices in priority queue, Q, in d[v]'s increasing order;
    while not Empty(Q) {
        u = ExtractCheapest(Q);
        S[u] = 1; /* Add u to S */

        for each vertex v adjacent to u
            if (S[v] ≠ 1 and d[v] > d[u] + w[u, v]) {
                remove v from Q;
                d[v] = d[u] + w[u, v];
                pi[v] = u;
                insert v into Q according to its d[v];
            }
    } // end of while loop
}
```

Initialize takes O(|V|)

putting |V| inside but possible to be |E| since edges will continue to be explored

|E|

O(log(E))

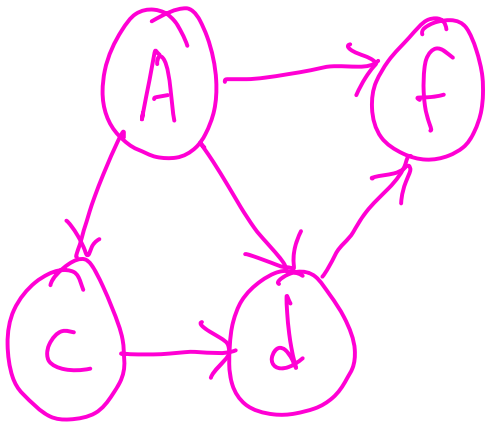
log cause it is binary heap

O(log(E))

$\therefore (|V| + |E| \log |E|)$

Initialize

Necessary Operation for this algorithm



|V| : 4

|E| : 5