4)

a) Modificaría el grafo creando una funcion costo que le atribuya un costo segun los nodos a cada arista

Costo (a , b):
    return a + b

de esta forma dijkstra puede aplicarse directamente
La complejidad de calcular el costo sería $O(E)$
y dijkstra es $O((V+E)\cdot\log(V))$ por lo tanto, la complejidad asintótica queda igual.

$$O(E) \in O((V+E)\cdot\log(V))$$

b) Un algoritmo que utilizaría, sería realizar dos BFS, uno desde U a X y el otro desde V a X, manteniendo registro de la distancia. Luego sumaría ambas distancias obtenidas para obtener el diametro del arbol. Esto funcionaría ya que sabemos que existe un solo camino de largo maximo y que X esta a una distancia maxima de U o V, por lo tanto el BFS conecta a ambos nodos en X.

c) PRIM_MODIFICADO(s, G) :

```
Q = cola de prioridades vacia
T = list vacia
for u ∈ V(G) - {s} :
    d[u] = ∞
    π[u] = ∅
d[s] = 0
π[s] = (∅, ∅)
Insert(Q, s)
while Q no esta vacia :
    u = Extract(Q)
    T = T ∪ {(π[u], u)}
    for v ∈ α[u] :
        if v ∈ Q :
            min = ∞
            for arista in aristas(u,v) :
                if wi < min :
                    min = wi
                    etiqueta = li
            if d[v] > min :
                d[v] = min
                π[v] = (u, etiqueta)
return T
```

se revisan las aristas entre u y v

se utiliza el costo minimo de cada conexion

se guarda la etiqueta con menor costo