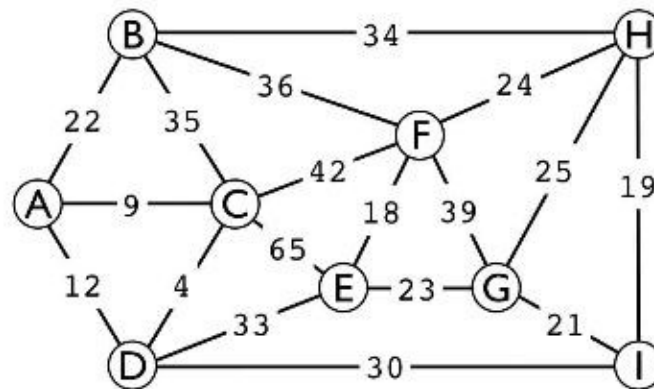


### Atividade Avaliativa 6 - Algoritmos gulosos (MST's)

1. No algoritmo de Kruskal, explique o que são e como funcionam as primitivas `Make_Set(v)`, `Find_Set(v)` e `Union(u, v)` descrevendo o pseudo-código de cada uma delas.
2. Execute o algoritmo de Kruskal no grafo a seguir, mostrando o trace completo (passo a passo). Qual é a MST obtida? Quanto vale seu peso?

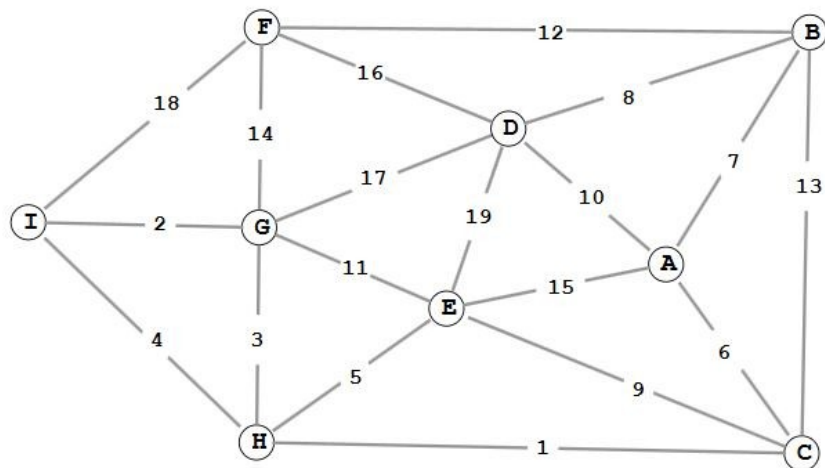


3. Mostre que o algoritmo de Kruskal é ótimo, ou seja, sempre retorna uma MST de  $G$ .
4. Mostre que a complexidade do algoritmo de Kruskal é  $O(m \log n)$ .
5. Explique o funcionamento do algoritmo de Prim. Porque ele é considerado um algoritmo guloso?

```

MST_Prim( $G, w, r$ ) {
    for each  $v \in V$  {
         $\lambda(v) = \infty$ 
         $\pi(v) = \text{nil}$ 
    }
     $\lambda(r) = 0$ 
     $Q = \emptyset$ 
    for each  $v \in V$ 
        Insert( $Q, v$ )
    while  $Q \neq \emptyset$  {
         $u = \text{ExtractMin}(Q)$ 
         $S = S \cup \{u\}$ 
        for each  $v$  in  $N(u)$  {
            if  $v \in Q$  and  $w(u,v) < \lambda(v)$  {
                 $\lambda(v) = w(u,v)$ 
                 $\pi(v) = u$ 
                Decrease_Key( $Q, v, w(u,v)$ )
            }
        }
    }
}
    
```

6. Execute o algoritmo de Prim no grafo a seguir com raiz no vértice I, mostrando o trace completo (passo a passo). Qual é a MST obtida? Quanto vale seu peso?



7. Demonstre a propriedade do corte, que relaciona um subconjunto  $S$  de  $V$ , a aresta de menor peso com uma extremidade em  $S$  e uma MST de  $G$ .

8. Prove a otimalidade (corretude) do algoritmo de Prim.

"Start by doing what's necessary; then do what's possible; and suddenly you're doing the impossible."  
-- São Francisco de Assis