

Algoritmo de Prim

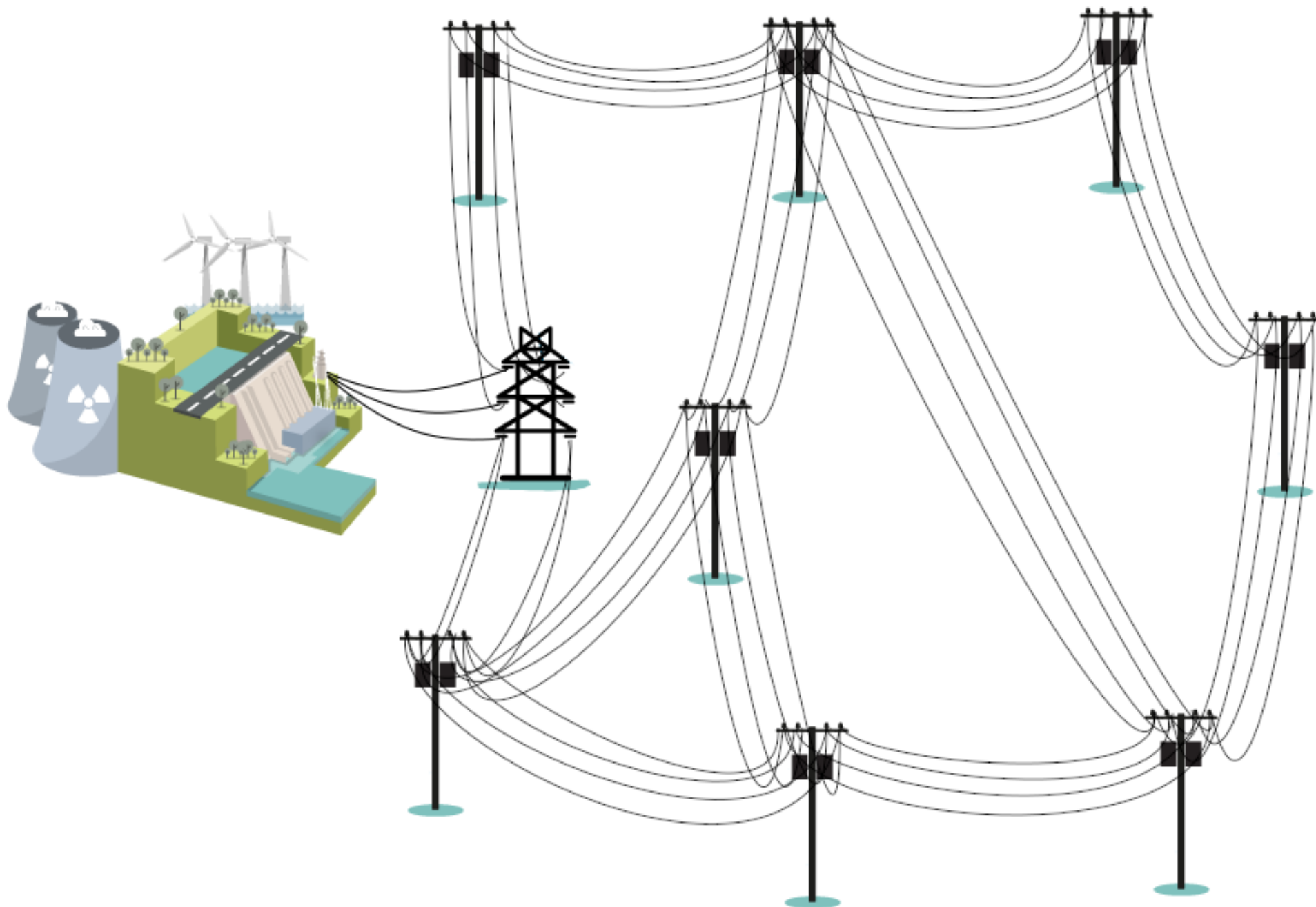
Eric dos Santos Coelho
Samuel Lima da Silva

<https://github.com/huffman-aline-eric-samuel/huffman>



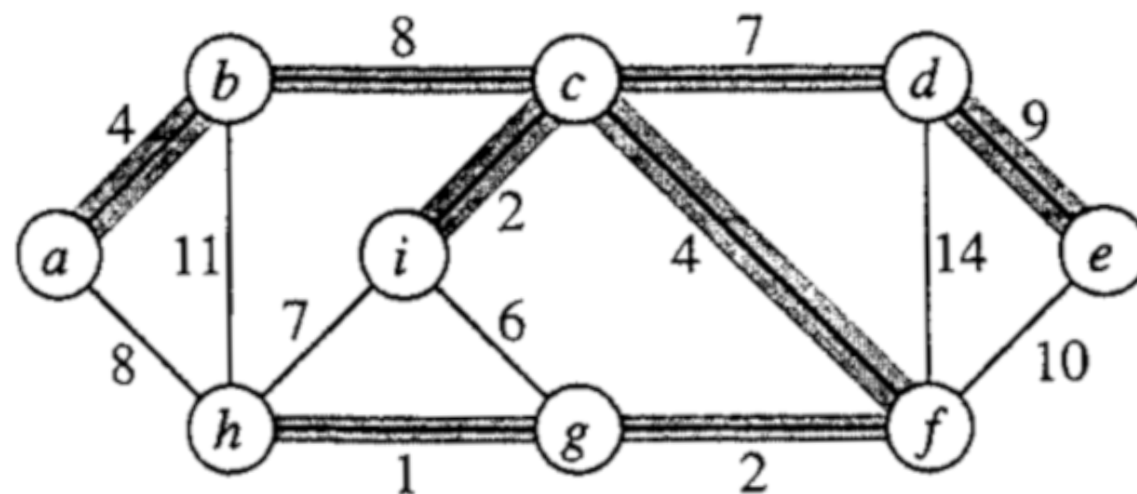
Motivação

Construir grafo com numero minimo de arestas que ligue todas os vertices e tenham o menor custo



Algoritmo

- Desenvolvido por Vojtěch Jarník e Robert Clay Prim e redescoberto por Edsger Wybe Dijkstra.
- O algoritmo de Prim é um algoritmo guloso
- Encontra uma árvore geradora mínima para um grafo



Definições

- Grafo conexo:
Um grafo é dito conexo se para quaisquer dois vértices do grafo existe um ou mais caminhos entre eles.
- Árvore geradora:
Dado um grafo conexo, não direcionado G , uma árvore geradora é todo subgrafo conexo de G que seja árvore.
- Árvore geradora mínima:
É uma árvore geradora de um grafo não direcionado, conexo e com pesos nas arestas tal que o custo total de suas arestas seja o menor possível.

Criação da Árvore

```
grafo* algoritmo_prim(grafo* g, int partida){
    grafo* ACM = criar_grafo(g->tamanho);
    heap* hp;
    adj_lista* arestas;
    adj_lista* min_aresta;
    int adicionados =1;
    arestas = g->vertices[partida];
    hp = criar_heap(100);
    while(adicionados < g->tamanho) {
        while(arestas != NULL) {
            enqueue(hp, arestas);
            arestas = arestas->prox;
        }
        min_aresta = dequeue(hp);
        if(ACM->vertices[min_aresta->vertice] == NULL) {
            adicionar_vertice(ACM, min_aresta->origem, min_aresta->vertice, min_aresta->custo);
            arestas = g->vertices[min_aresta->vertice];
            adicionados++;
        }
    }
    return ACM;
}
```