Exercícios de Parametrizada

Matheus Souza D'Andrea Alves

2018.2

Bounded search tree

Cluster editing

Observe as seguintes definições:

Definição 1: Grafo Cluster

Um grafo G onde toda componente conexa δ_i(G) é uma clique.

Definição 2: Grafo livre de P_3

Um grafo G é chamado de livre de P_3 se e somente se não possuir um grafo caminho com 3 vértices.

Dessa forma demonstraremos o seguinte teorema.

Teorema 1: Um grafo G é cluster se e somente se é livre de P_3.

Demonstração.
Suponha por absurdo que G não é livre de P_3 isso implica em que exista um subgrafo induzido G' onde existem dois vértices não vizinhos na mesma componente, isso implica que alguma componente de G não é uma clique o que é absurdo.

Suponha agora que um grafo H qualquer não possua P_3 isso implica em que dado quaisquer pares de vértices $u, v \in V(G)$ ou $\exists(u, v) \in E(G)$, ou os vértices estão em componentes distintas de H, e portanto H é um cluster. $\square$

Abordaremos o problema de cluster editing para esse exercício conforme descrito abaixo.

Problema 1: Cluster editing

Entrada: *Um grafo G*
Questão: Qual é o menor número de arestas que podem ser removidas ou inseridas em G que causam o grafo resultante ser um cluster.

Como vimos acima, um grafo cluster é livre de P_3 e portanto podemos abordar o problema de cluster editing como o problema de eliminação de P_3 em um grafo G. Para eliminar um p3 existem 3 possibilidades:

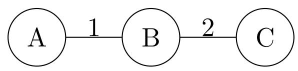

Figura 1: Um P3

- Remover aresta 1
- Remover aresta 2
- Adicionar aresta entre A e C

Sabemos que é possível encontrar P_3 em tempo $\mathcal{O}(n+m)$ portanto usando o número de movimentos restantes como paramêtro segue o algoritmo.

```
Function ClusterEditable(Graph g, int remainingMovements): bool{
  if g.hasP3()  {
    if remainingMovements = 0 {
      return false
    } else {
      p3 = g.getP3()
      g1 = g.removeEdge(p3.edges[0])
      g2 = g.removeEdge(p3.edges[1])
      g3 = g.addEdge(p3.vertices[0],p3.vertices[2])
      remainingMovements--
      return ClusterEditable(g1,remainingMovements)
            or ClusterEditable(g2,remainingMovements)
            or ClusterEditable(g3,remainingMovements)
    }
  } else
      return true
}
```

Tal algoritmo é resolvível em tempo $\mathcal{O}(k^3(n+m))$

Distância entre Strings