Relatório 2º projeto ASA 2022/2023

Grupo: AL086

Aluno(s): Afonso Azaruja(103624)

Descrição do Problema e da Solução

Pretende-se encontrar o caminho num grafo, não direcionado e pesado, que maximize o peso (trocas comerciais no contexto do problema).

Para resolver este problema foi utilizado o algoritmo kruskal com o intuito de encontrar uma *maximum spanning tree*. É utilizado um vetor com todos os arcos, ordenados de forma decrescente de acordo com o peso, por sua vez os arcos escolhidos são sempre os de maior peso mas com a condição de que este não forme um ciclo com a *spanning tree* que está a ser formada, em seguida é adicionado o valor do peso desse arco para o total.

Análise Teórica

```
main() {
    ParseVE() O(1)
    DisjSet obj(V) O(V)
    vector<Edge> edges(E) O(E)
    for (i < E) O(E)
        ParseEdges()
        initializeEdge()
    endfor
    SortEdges() O(E*logE)
    for (Edge e : edges) O(E*logE)
        if (find(e.u) != find(e.v))
            union(e.u, e.v)
            res += e.w
        endif
    endfor
}
```

```
makeSet() { O(V)
    for (i < n)
        parent[i] = i
    endfor
}
find(int x) { O(E)
    if (parent[x] != x)
        parent[x] = find(parent[x])
    endif
    return parent[x]
}
unionSet(int x, int y) { O(2*E)
    int xp = find(int x)
    int yp = find(int y)
    … (const)
}
```

- Leitura dos dados de entrada: **O(E)**
- Inicialização DisjSet, com o makeSet(): **O(V)**
- Ordenação do vetor de arcos: **O(E*logE)**
- Algoritmo kruskal: **O(E*logE + V + E*logE + E*logE) = O(E*logE)**

Complexidade global da solução: **O(E) + O(V) + O(E*logE) + O(E*logE) = O(E*logE)**

Relatório 2º projeto ASA 2022/2023

Grupo: AL086

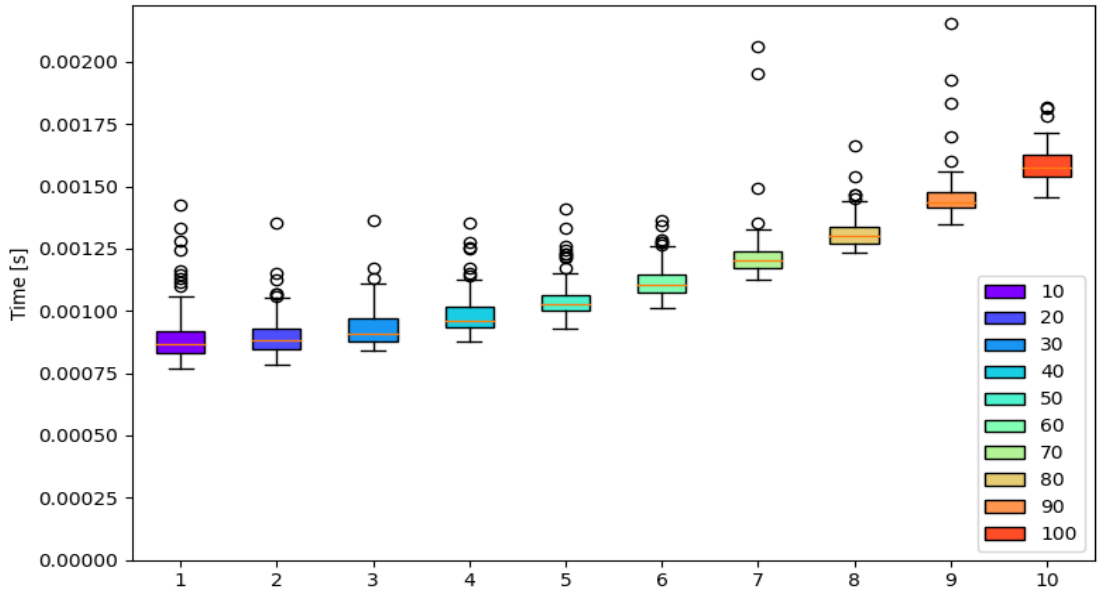
Aluno(s): Afonso Azaruja(103624)

Avaliação Experimental dos Resultados

Foram criadas 10 instâncias de input usando a ferramenta *dgg.c* disponibilizada que gera grafos completamente ligados. Os testes iniciam-se com um grafo de 10 vértices e em cada iteração a ordem de grandeza é incrementada em 10, ou seja, o 2º teste terá 20 vértices e o 3º 30 e.t.c.

A ferramenta utilizada para a execução dos testes foi o *hyperfine*, no qual os comandos introduzidos foram:

- --warmup 2
- --runs 100



Ao analisar o gráfico conclui-se que este está concordante com a análise teórica realizada previamente.