**Computador:** Notebook CCE Aspire E 15 E5-574g-75ME

**Sistema Operacional**: Linux Ubuntu 16.04

**Processador**: Intel Core i7, 6° geração

**Ram**: 8 Gb

**Dupla**: Francisco Ferreira da Silva Junior (377588) e Maria Juliana Pontes Mateus (373703)

**Resultados Totais**

|  |
| --- |
| Linkset e Findset eficientes |
|  | 0 | 1 | 2 | 3 |
| 60 | 8 | 1 | 2 | 1 |
| 100 | 2 | 3 | 2 | 1 |
| 200 | 9 | 7 | 7 | 8 |
| 300 | 16 | 13 | 21 | 16 |
| 400 | 47 | 43 | 38 | 45 |
| 500 | 80 | 106 | 88 | 89 |
| 600 | 161 | 172 | 148 | 135 |
| 700 | 291 | 262 | 262 | 233 |
| 800 | 409 | 366 | 388 | 365 |
| 900 | 635 | 663 | 678 | 603 |

|  |
| --- |
| Linkset e Findset ineficientes |
|  | 0 | 1 | 2 | 3 |
| 60 | 8 | 3 | 6 | 1 |
| 100 | 6 | 3 | 5 | 5 |
| 200 | 6 | 8 | 8 | 12 |
| 300 | 25 | 19 | 27 | 31 |
| 400 | 62 | 67 | 55 | 58 |
| 500 | 99 | 151 | 158 | 120 |
| 600 | 210 | 216 | 187 | 232 |
| 700 | 372 | 348 | 322 | 315 |
| 800 | 497 | 645 | 600 | 495 |
| 900 | 844 | 808 | 830 | 685 |

**Com a construção da tabela de tempos para cada teste de instância, podemos notar o que já era previsto, com dois tipos de métodos, para Find\_set e Link\_set, sendo um mais eficiente que o outro, usando o algoritmo Kruskal com os dois métodos eficientes da classe ConjuntoDisjunto, notamos que com o uso dos métodos eficientes, é mais rápido em todos os testes a criação de grafos, com diversas instancias de tamanho.**

|  |
| --- |
| Linkset eficiente e Findset ineficiente |
|  | 0 | 1 | 2 | 3 |
| 60 | 5 | 1 | 3 | 4 |
| 100 | 5 | 2 | 2 | 3 |
| 200 | 7 | 6 | 8 | 8 |
| 300 | 18 | 20 | 27 | 23 |
| 400 | 46 | 66 | 42 | 62 |
| 500 | 119 | 125 | 138 | 110 |
| 600 | 319 | 302 | 298 | 172 |
| 700 | 317 | 323 | 388 | 305 |
| 800 | 447 | 437 | 504 | 460 |
| 900 | 743 | 795 | 771 | 665 |

|  |
| --- |
| Linkset ineficiente e Findset eficiente |
|  | 0 | 1 | 2 | 3 |
| 60 | 6 | 3 | 2 | 1 |
| 100 | 3 | 4 | 2 | 2 |
| 200 | 6 | 9 | 5 | 8 |
| 300 | 17 | 15 | 22 | 17 |
| 400 | 49 | 49 | 36 | 48 |
| 500 | 82 | 115 | 99 | 111 |
| 600 | 188 | 239 | 363 | 197 |
| 700 | 321 | 303 | 310 | 262 |
| 800 | 481 | 392 | 446 | 413 |
| 900 | 711 | 694 | 747 | 621 |

**Agora alternando entre um teste com Linkset eficiente e Findset ineficiente e outro teste com Linkset ineficiente e Findset eficiente podemos ver que se destaca mais o teste usado com Linkset ineficiente e Findset eficiente sendo mais rápido na maioria das instancias, mas sem ser mais rápido que o teste feito com o Findset e o Linkset eficientes.**

**Com essa atividade podemos concluir que o algoritmo Kruskal é bem mais eficiente tanto para pequenas quanto para grandes entras de dados, usando métodos eficientes do ConjuntoDisjunto, também trabalhando com o Quick\_Sort melhorando mais ainda sua eficácia.**