Joaquín Lezama

4.508.033/0

Informe

A continuación se presentan los resultados de medir los tiempos de las operaciones sobre matrices.

3) Predicados sobre matrices

|  |  |  |
| --- | --- | --- |
| **(i,j)** | **interferences** | **time** |
| (5,5) | 412 | 0,000 s |
| (20,20) | 18.142 | 0,016 s |
| (50,50) | 262.852 | 0,144 s |
| (100,100) | 2.050.702 | 1,026 s |
| (200,200) | 16.201.402 | 12,359 s |
| (500,500) | *Out of global stack* | *Out of global stack* |

4) Predicados eficientes sobre matrices

|  |  |  |
| --- | --- | --- |
| **(i,j)** | **interferences** | **time** |
| (5,5) | 883 | 0,000 s |
| (20,20) | 3.363 | 0,001 s |
| (50,50) | 20.403 | 0,007 s |
| (100,100) | 80.803 | 0,037 s |
| (200,200) | 321.603 | 0,118 s |
| (500,500) | 2.004.004 | 0,785 s |

Conclusión

Se puede observar en los resultados que las operaciones implementadas en la parte 4) son mucho más óptimas que las implementadas en la parte 3). Esto resulta evidente ya que en la parte 3) se trabajaron las matrices como listas de listas, por lo que cada operación requiere recorrer varias veces estas listas. En la parte 4) al definirse las matrices como functors el acceso a los elementos de esta se hace directo.