

Eficiencia empírica de **algoritmos**

Códigos

```
1  /**
2   @file Cálculo de la sucesión de Fibonacci
3   */
4  #include <iostream>
5  using namespace std;
6  #include <ctime>
7  #include <chrono>
8  /**
9   @brief Calcula el término n-ésimo de la sucesión de Fibonacci.
10
11   @param n: número de orden del término buscado. n >= 1.
12
13   @return: término n-ésimo de la sucesión de Fibonacci.
14   */
15
16  using namespace std::chrono;
17
18  int fibo(int n)
19  {
20      if (n < 2)
21          return 1;
22      else
23          return fibo(n-1) + fibo(n-2);
24  }
25
26  int main(int argc, char* argv[])
27  {
28      high_resolution_clock::time_point t_antes, t_despues;
29      int n;
30      int f;
31
32      //cout << "Numero del termino: ";
33      //cin >> n;
34      n=atoi(argv[1]);
35
36      t_antes = high_resolution_clock::now();
37      f = fibo(n);
38      t_despues = high_resolution_clock::now();
39
40      //cout << "El término " << n << "-ésimo es: " << f << endl;
41      duration<double> transcurrido = duration_cast<duration<double>>(t_despues-t_antes);
42      cout << n << " " << transcurrido.count() << "\n";
43
44      return 0;
45  }
```

```
1  #####
2  # Makefile
3  #####
4
5  STUDDATA = $(wildcard data/*/*.dat)
6  STUDGRPH = $(wildcard grphx/*/*.{jpg,png})
7
8  BIN = bin
9
10 SRC = $(wildcard src/*.cpp)
11 EXE = $(basename $(SRC))
12 ALG = $(wildcard *.cpp)
13
14 CXXFLAGS = -std=gnu++0x
15
16 default: $(EXE)
17
18 %: %.cpp
19     $(CXX) $(CXXFLAGS) $< -o $(patsubst src%,bin%, $(basename $< ))
20
21 clean:
22     $(RM) -v $(BIN)/*
23
24 mrproper: clean
25     $(RM) -v $(BIN)/* $(STUDDATA) $(STUDGRPH)
26
27 .PHONY: clean
28 .PRECIOUS: $(LOG)
29 .NOEXPORT:
30
31 #####
32
```

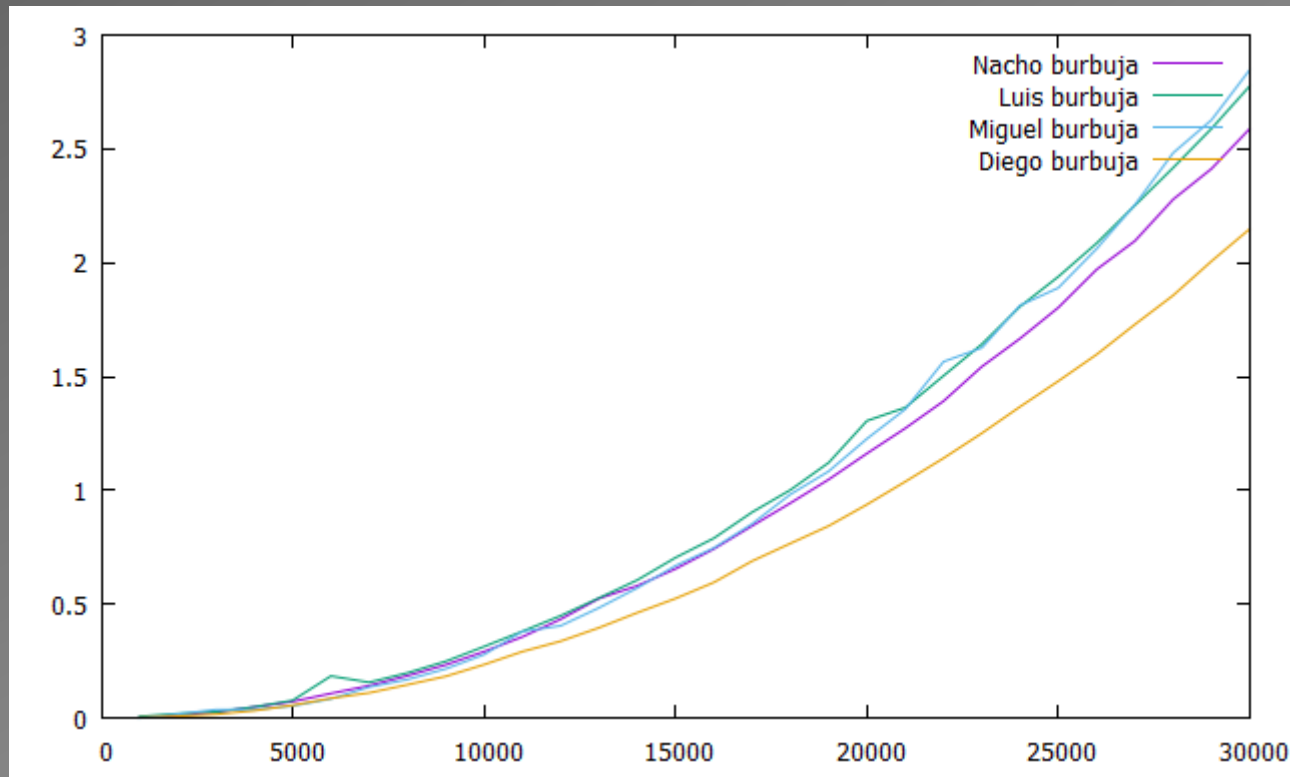
Códigos

```
1 #!/bin/bash
2 #Burbuja
3
4 i=10000
5 echo -e "Ejecutando burbuja con tamaño maximo 30.000 a saltos de 1.000"
6 sleep 3
7 while [ $i -le 30000 ]
8 do
9     ./bin/burbuja $i >> ./data/$1/burbuja.dat
10    echo -e "Iteración $i"
11    ((i+=1000))
12 done
13
14 #Fibonacci
15
16 i=1
17 echo -e "Ejecutando fibonacci con tamaño maximo 45 a saltos de 1"
18 sleep 3
19 while [ $i -le 45 ]
20 do
21     ./bin/fibonacci $i >> ./data/$1/fibonacci.dat
22     echo -e "Iteración $i"
23     ((i+=1))
24 done
25
26 #heapsort
27
28 i=10000
29 echo -e "Ejecutando heapsort con tamaño maximo 1.000.000 a saltos de 10.000"
30 sleep 3
31 while [ $i -le 1000000 ]
32 do
33     ./bin/heapsort $i >> ./data/$1/heapsort.dat
34     echo -e "Iteración $i"
35     ((i+=10000))
36 done
37
38 #insercion
39
40 i=1000
41 echo -e "Ejecutando insercion con tamaño maximo 30.000 a saltos de 1.000"
42 sleep 3
43 while [ $i -le 30000 ]
44 do
45     ./bin/insercion $i >> ./data/$1/insercion.dat
46     echo -e "Iteración $i"
47     ((i+=1000))
```

```
1 #!/usr/bin/python
2
3 import os
4 import sys
5 import re
6 import numpy as np
7 import matplotlib.pyplot as plt
8
9 path= './data/' + str(sys.argv[1]) + '/'
10
11 for filename in os.listdir(path):
12
13     with open(path+filename) as f:
14         data = f.read()
15
16         data = data.split('\n')
17
18         data = [row for row in data if row != ""]
19         x = [row.split(' ')[0] for row in data]
20         y = [row.split(' ')[1] for row in data]
21
22         fig = plt.figure()
23
24         ax1 = fig.add_subplot(111)
25
26
27         title = filename.replace(".dat","")
28
29         ax1.set_title(title)
30         ax1.set_xlabel('Tiempo')
31         ax1.set_ylabel('Ejecuciones')
32
33         ax1.plot(x,y, c='r', label=title)
34
35
36         savepath = path.replace("data","grphx")
37
38         d = os.path.dirname(savepath+title+".png")
39         if not os.path.exists(d):
40             os.makedirs(d)
41
42         plt.savefig(savepath+title+".png")
```

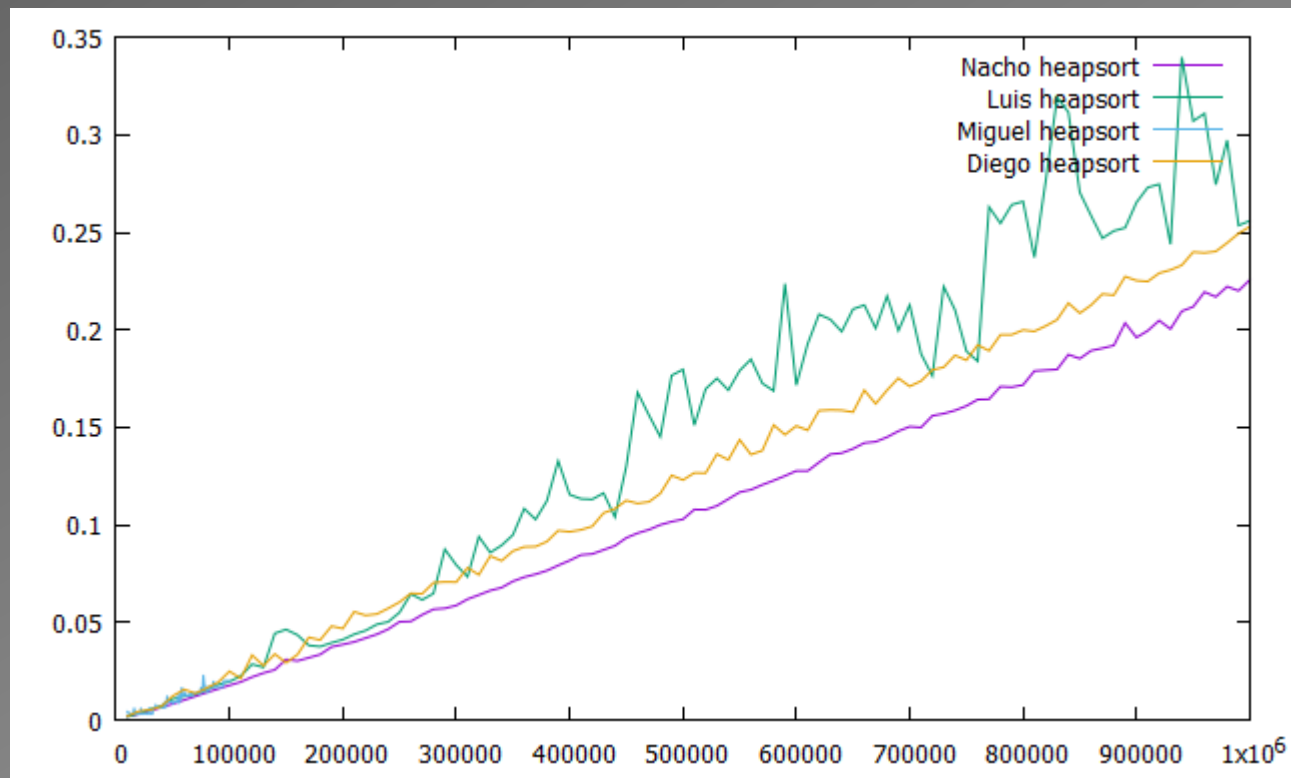
Algoritmos de ordenación

Burbuja



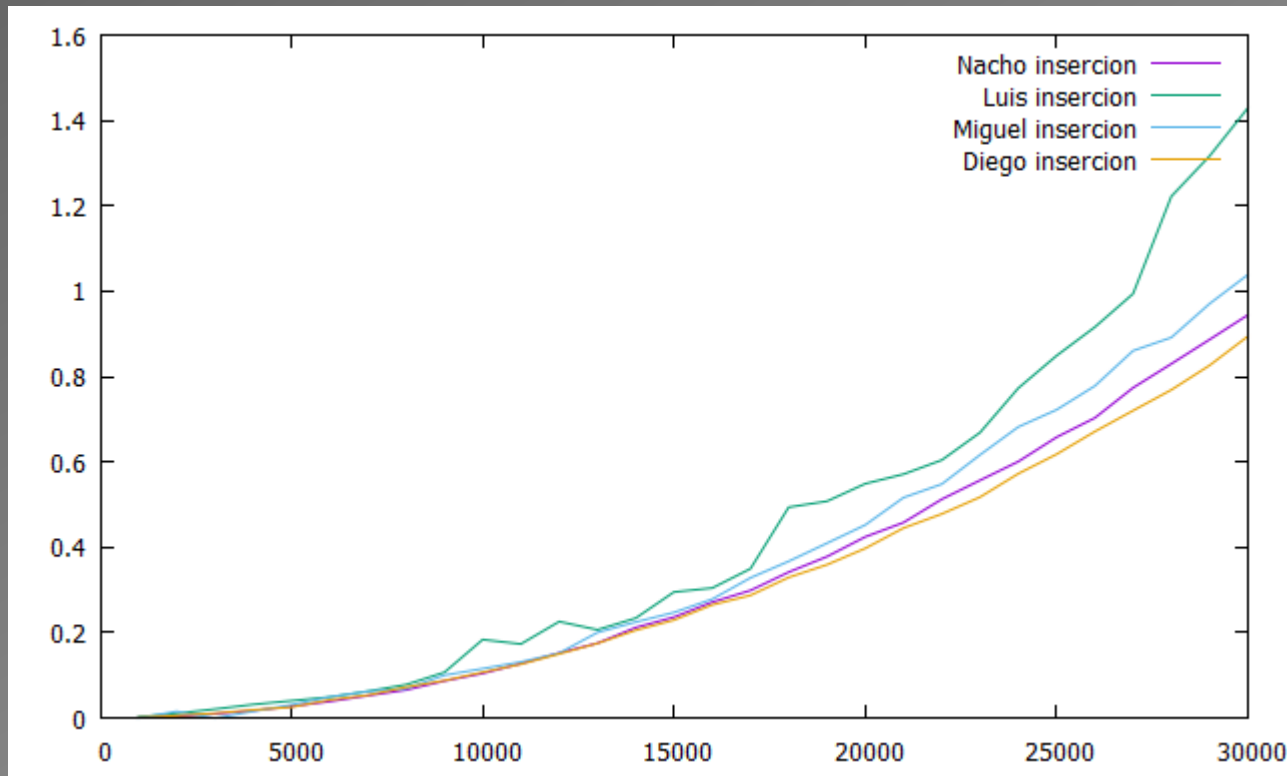
Algoritmos de ordenación

Heapsort



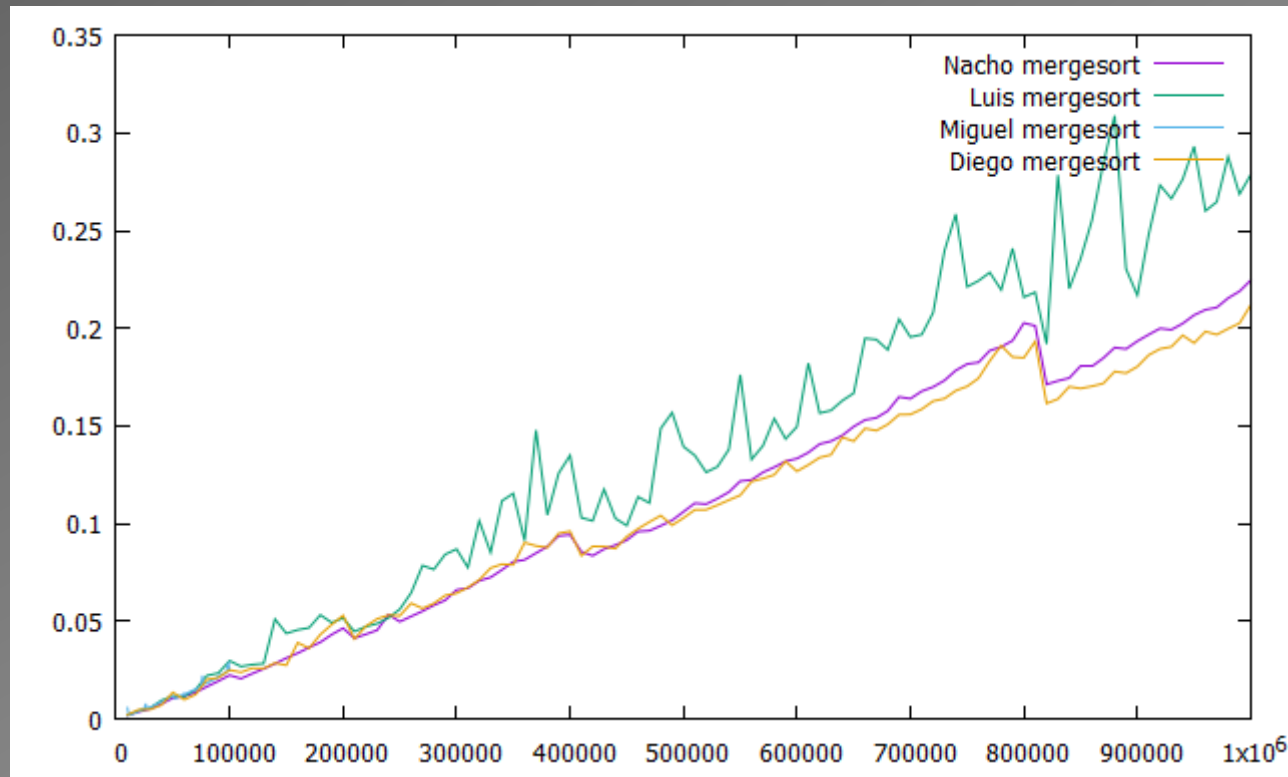
Algoritmos de ordenación

Inserción



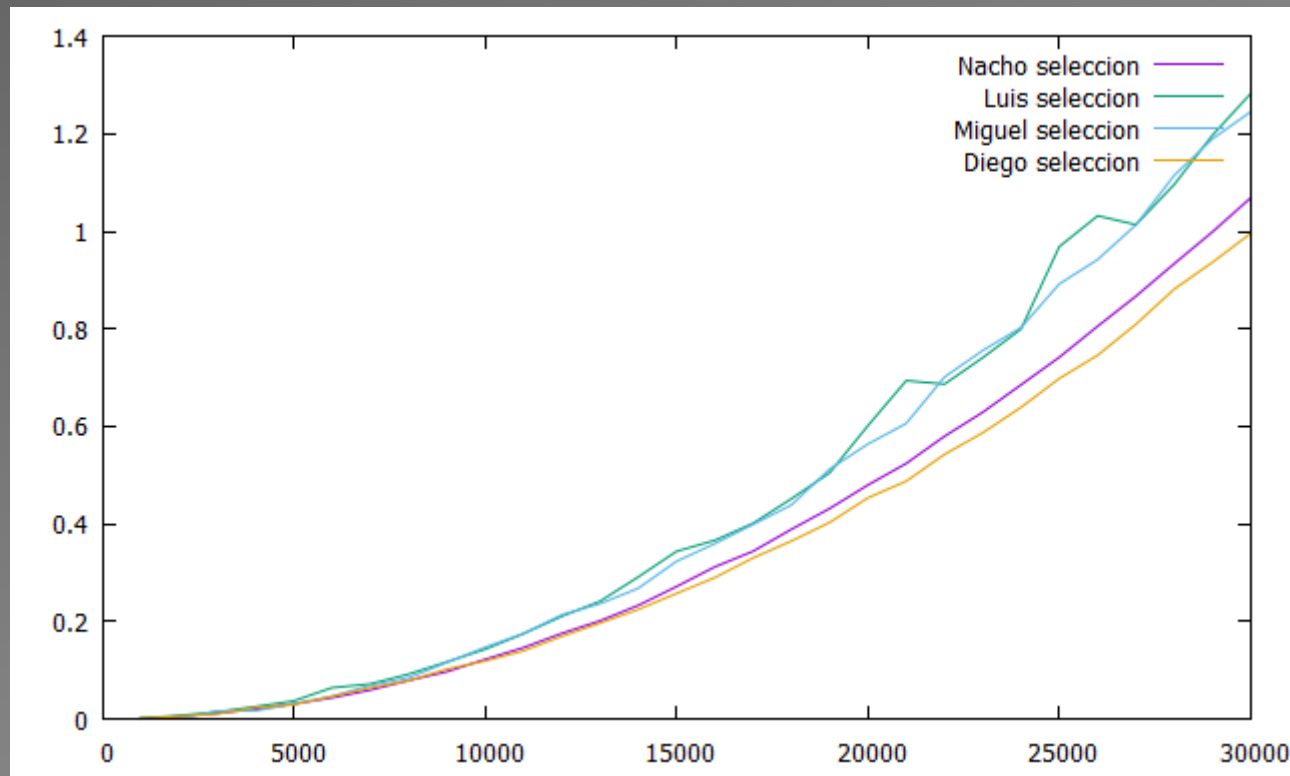
Algoritmos de ordenación

Mergesort

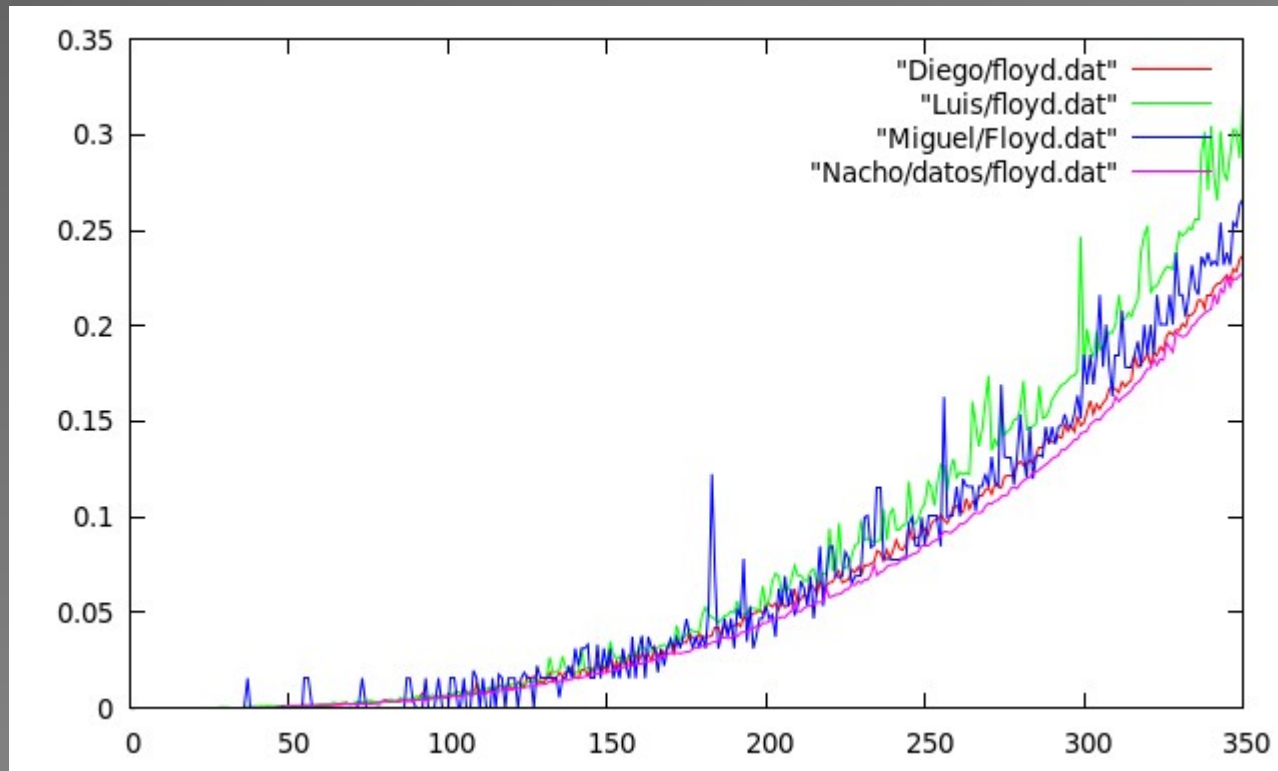


Algoritmos de ordenación

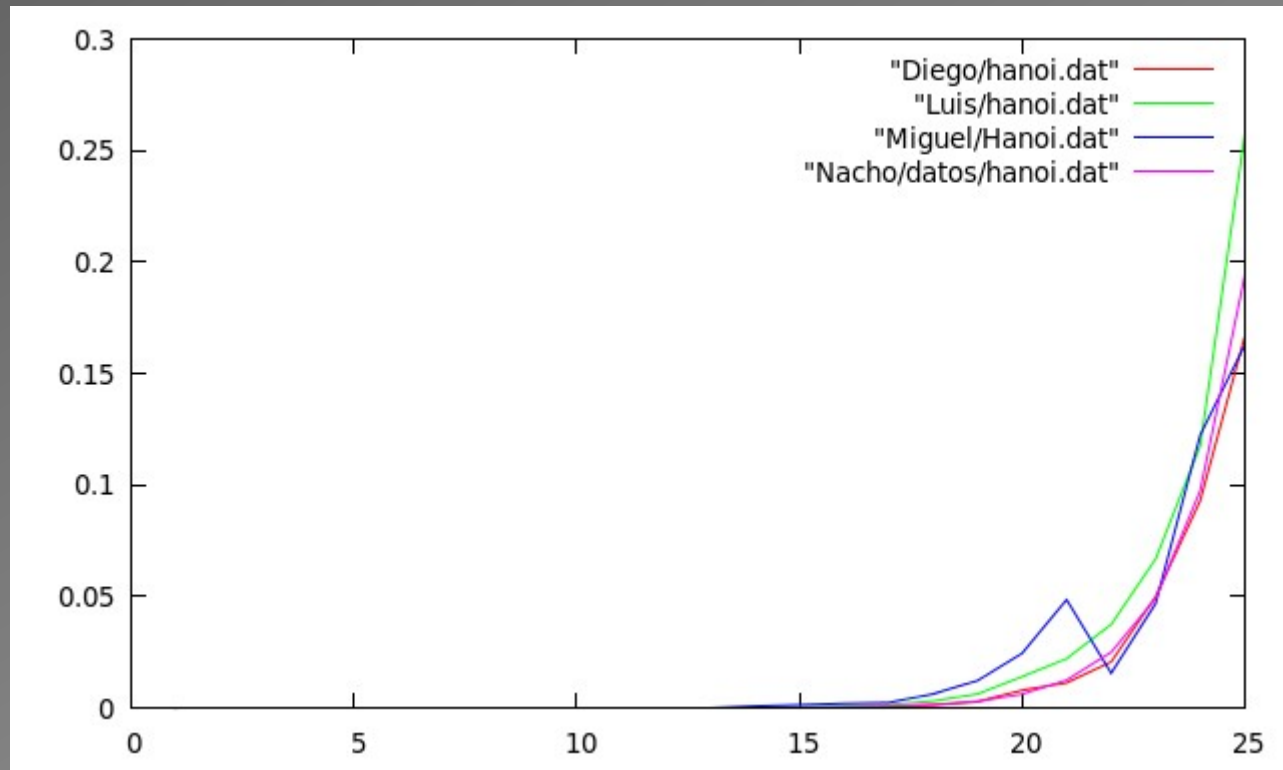
Selección



Algoritmo de Floyd



Algoritmo de Hanoi



Algoritmo de Fibonacci

