

# Proyecto 01

Abner Iván García Alegría 21285

Oscar Esteban Donis Martínez 21610

Astrid Marie Glauser Oliva 21299

Gonzalo Enrique Santizo vega 21504

Jose Daniel Gomez Cabrera 21429

# Repositorio de Github

<https://github.com/estebandonis/Proyecto1-AnalisisdeAlgoritmos>

estebandonis

Proyecto1-AnalisisdeAlgoritmos

Type to search

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Proyecto1-AnalisisdeAlgoritmos

Private

Watch 1

Fork 0

Star 0

main 1 Branch 0 Tags

Go to file

Add file

<> Code

GonzaloSantizo

final changes

7a1f085 · 4 days ago

5 Commits

.vs

final changes

4 days ago

machine.txt

Maquinna de Turing de la secuencia de Fibonacci implement...

5 days ago

main.py

final changes

4 days ago

README

Add a README

Help people interested in this repository understand your project by adding a README.

Add a README

About

No description, website, or topics provided.

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Contributors 2

estebandonis

Esteban Donis

GonzaloSantizo

Languages

Python 100.0%

Suggested workflows

Based on your tech stack

# machine.txt

```
1 ESTADOS = {q0, q1, q2, q3, q4, q5, q6, q7, q8, q9, q10, q11, q12, q13, q14, q15, q16, q17, q18, q19, q20, q21, q22, q23, q24, q25, q26, q27, q28, q29, q30, q31}
2 ALFABETO DE ENTRADA = {}
3 ALFABETO DE CINTA = { |, 1, X, Y, B}
4 FUNCION DE TRANSICION = {
5     (q0, B) → (q0, B, R),
6     (q0, |) → (q1, B, R),
7     (q1, B) → (q5, |, R),
8     (q1, |) → (q2, |, R),
9     (q2, B) → (q4, B, R),
10    (q2, |) → (q3, |, R),
11    (q3, |) → (q3, |, R),
12    (q3, X) → (q3, X, R),
13    (q3, B) → (q6, X, R),
14    (q6, B) → (q7, |, L),
15    (q7, X) → (q8, X, L),
16    (q8, |) → (q8, |, L),
17    (q8, B) → (q9, B, R),
18    (q8, X) → (q17, X, L),
19    (q9, |) → (q3, B, R),
20    ...
```

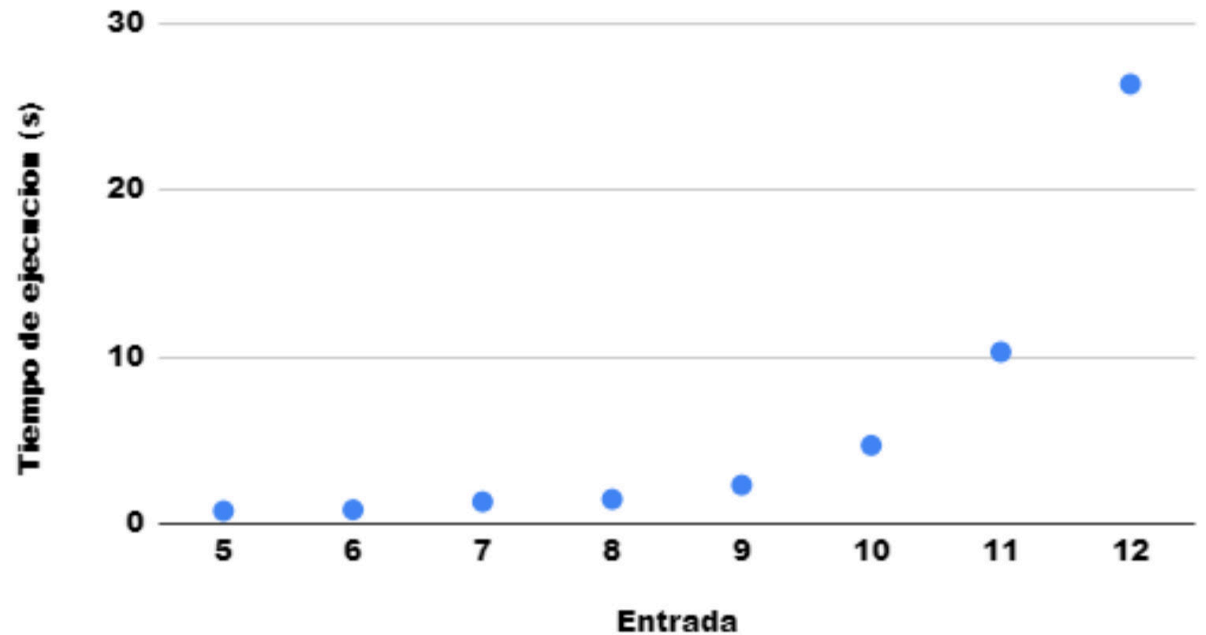
# main.py

```
1 def main():
2     # Leemos los datos del archivo
3     estados, alfabeto_entrada, alfabeto_cinta, func_transition, estado_inicial, sim_blanco, estados_aceptacion = read_file()
4
5     # Definimos la secuencia de Fibonacci como entrada
6     input_sequence = int(input("Ingrese el numero a averiguar en la secuencia de Fibonacci: "))
7     numCinta = ""
8     i = 0
9     while i < input_sequence:
10         numCinta += '|'
11         i += 1
12     # Convertimos la entrada en una cinta con un número definido de blanks
13     blank_number = 100
14     w = list(sim_blanco * blank_number + numCinta + sim_blanco * (blank_number*4))
15
16     # Le pasamos los datos de la máquina de turing a la función para ejecutarla
17     maquina_turing(estados, alfabeto_entrada, alfabeto_cinta, func_transition, estado_inicial, sim_blanco, estados_aceptacion, w, blank_number)
18     print("Número ingresado: ", input_sequence)
19
20 if __name__ == "__main__":
21     start_time = time.time() # Start measuring time
22     main() # Call the main function
23     end_time = time.time() # Stop measuring time
24     execution_time = end_time - start_time # Calculate execution time
25     print("Execution time:", execution_time, "seconds")
```

Entrada	Tiempo de ejecución en segundos
5	0.78
6	0.85
7	1.33
8	1.48
9	2.33
10	4.71
11	10.31
12	26.37

# Diagrama de Dispersión

Tiempo de ejecucion (s) frente a Entrada



El algoritmo clásico para generar la secuencia de Fibonacci es iterativo, sumando los dos números anteriores de la secuencia para obtener el siguiente. A medida que la secuencia avanza, el tiempo necesario para calcular cada nuevo término aumenta, debido al crecimiento rápido de los números en la secuencia. Esto puede resultar en un crecimiento que parece exponencial en una gráfica que compare la entrada (el índice  $n$  de la secuencia) con el tiempo de cálculo necesario.

# Regresión polinomial que se ajuste mejor a los datos



## Tiempo de ejecución

$$2.96 - 4.01x + 0.976x^2$$

Donde "x" representa algún valor o parámetro relacionado con la entrada o el tamaño del problema. La ecuación muestra cómo el tiempo de ejecución varía a medida que "x" cambia.

$$R^2 = 0.91$$

El coeficiente  $R^2$  es una medida de cuán bien se ajusta el modelo a los datos reales. Un valor de  $R^2$  de 0.91 indica que el modelo explica aproximadamente el 91% de la variabilidad en los datos observados. En otras palabras, el modelo tiene un buen ajuste a los datos en este caso particular.



