busqueda_informada_p5 > busqueda_informada5.py > simulated_annealing

```python
import math
import random

def himmelblau(x,y):
    return (x**2 + y - 11)**2 + (x + y**2 - 7)**2

def simulated_annealing():
    t=10
    t_MIN=1e-3
    v_enfriamiento=100
    x_max, x_min= -5, 5
    y_max, y_min= -5, 5
    x_actual= random.uniform(x_max, x_min)
    y_actual= random.uniform(y_max, y_min)

    while t > t_MIN:
        for i in range(v_enfriamiento):
            x_nueva= x_actual +  random.uniform(-0.1, 0.1)
            y_nueva= y_actual +  random.uniform(-0.1, 0.1)
            if himmelblau(x_nueva, y_nueva) < himmelblau(x_actual, y_actual):
                x_actual, y_actual= x_nueva, y_nueva
            else:
                if random.random() < math.exp(-(himmelblau(x_nueva, y_nueva)- himmelblau(x_actual, y_actual))/t):
                    x_actual, y_actual= x_nueva, y_nueva

        t-=0.005
    return x_actual, y_actual, himmelblau(x_actual, y_actual)

mejor_x, mejor_y, resultado_min= simulated_annealing()

print(f"Mejor x: {mejor_x} \nMejor y: {mejor_y} \nResultado minimo: {resultado_min}")

```

Utilice las bibliotecas de math para usar el exponencial de e y random para asignarles valores a x, y.

Se definió una función himmelbalu para hacer la operación de esta.

Hacemos la función simulated_annealing donde es el punto principal del programa en ella se ejecutara el código del recocido simulado dentro de ella declaro la temperatura que es t, la temperatura mínima que es t_MIN, la velocidad de enfriamiento que es v_enfriamiento, asigno a x_actual un valor random dentro de mi rango que va 5 a -5 y hago lo mismo en y_actual

Comenzamos con el while que se ejecutara mientras t sea mayor a t_MIN y hacemos un ciclo for que tendrá el rango de la v_enfriamiento.

Dentro del for se buscan nuevos vecinos de nuestras x_actual, y_actual en este caso cada vecino puede sumarle o restarle 0.1

Pasamos a un if donde se evalua himmelblau con x_nueva, y_nueva debe ser menor a la evaluación de himmelblau con x_actual, y_actual de ser así x_actual, y_actual se le asignan los valores de x_nueva, y_nueva.

De no cumplirse el anterior pasamos al otro if donde se busca un numero randon entre 0 y 1 y si es menor a la función de probabilidad usada también se hará el cambio de x_actual, y_actual con los valores nuevos de x_nueva, y_nueva.

Se baja la temperatura en este caso de forma lineal y al final retornamos la mejor x, y y el resultado usando los mejore valores y al final imprimimos el resultado.

Temperatura usada 10

```
da_informada_p3/busqueda_informada3.py
Mejor x: -3.77428783560928
Mejor y: -3.2804715676701406
Resultado minimo: 0.00140434254506142 33
```

Temperatura usada 100

```
da_informada_p3/busqueda_informada3.py
Mejor x: -2.7932443052979856
Mejor y: 3.1311577354542
Resultado minimo: 0.0045582662992367 15
```

Temperatura usada 1000

```
da_informada_p3/busqueda_informada3.py
Mejor x: -2.798770420485257
Mejor y: 3.1357651625318153
Resultado minimo: 0.0021416401397246 02
```

Temperatura usada 10000

```
da_informada_p3/busqueda_informada3.py
Mejor x: 2.9997931370621913
Mejor y: 2.0001903593678576
Resultado minimo: 1.4117221880044744e-06
```