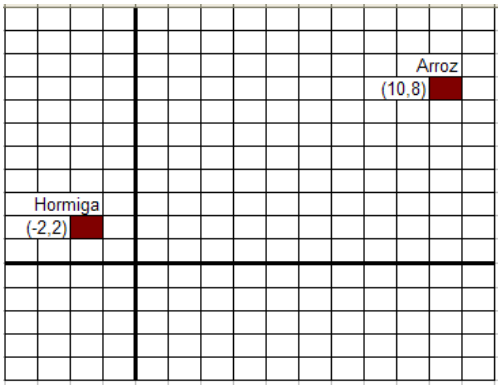
1ra Evaluación II Término 2007-2008. Diciembre 04, 2007

Tema 1 (30 puntos) En un plano cartesiano se encuentran una hormiga y un grano de arroz.
En cada instante de tiempo, la hormiga de manera aleatoria intuye la dirección donde ir (arriba, abajo, derecha, izquierda) y cuantas unidades desplazarse (entre 1 a 3) en la anterior dirección.



Implemente un algoritmo que simule 100 instantes de tiempo con desplazamientos de la hormiga que inicialmente se encuentra en las coordenadas (-2,2) y un grano de arroz en las coordenadas (10,8)
Al final indique las respuestas a las siguientes preguntas:

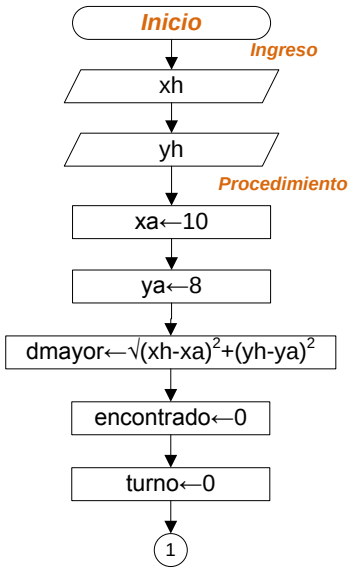
a. ¿La hormiga llegó al grano de arroz?
b. Si la respuesta a la pregunta anterior es "Si", entonces mostrar: cuántos pasos fueron necesarios.
c. ¿La distancia más lejana en la que estuvo la hormiga del grano de arroz?

$$d(P_1,P_2)=\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}$$

Rúbrica: Manejo de aleatorios (5 puntos), Control de movimiento y a) (10 puntos), b) (5 puntos) y c (5 puntos). Solución integral (5 puntos)

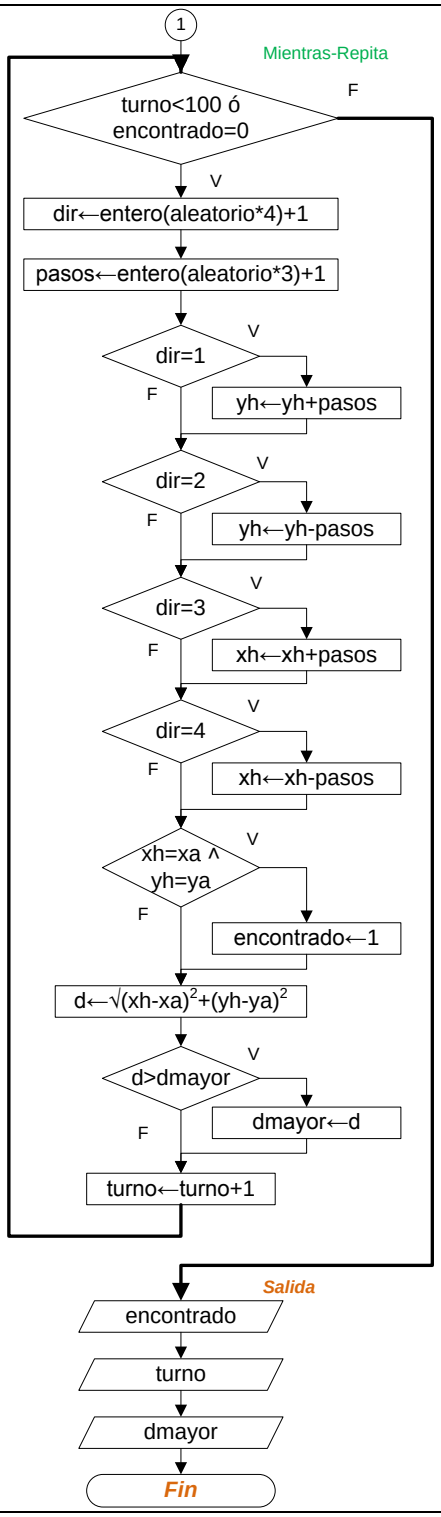
Propuesta de Solución:
Considere ingresar la ubicación inicial (**xh**,**yh**) de la hormiga y que la posición del arroz (**xa**,**ya**) sea fija.
Suponga que la distancia inicial es la mayor y que la hormiga no ha encontrado el grano de arroz.
Para la **dir**ección del movimiento de la hormiga y la cantidad de **pasos** se generan números aleatorios cuyos valores se usan para simular el movimiento al cambiar las coordenadas de la hormiga.
Luego de cada movimiento, se revisa si la hormiga encontró el grano de arroz o que las coordenadas sean iguales; también se puede revisar si la nueva distancia es mayor a las anteriores. Cuente un turno completado, y repita el procedimiento hasta que se completen los 100 turnos o se haya encontrado el grano de arroz.
Al final muestre los resultados buscados.
Tarea: Realizar las modificaciones para mostrar la cantidad de pasos en lugar de turnos.

Descripción	DIAGRAMA DE FLUJO	Python

Inicio
 	
Coordenada x de la hormiga
 	
Coordenada y de la hormiga
 	
Ubicación de grano de arroz en x
Ubicación del grano de arroz en y
 	
La distancia inicial es mayor
 	
No se ha encontrado todavía el grano de arroz
 	
Contador de turnos
		# ICM00794-Fund. de Computación - FCNM-ESPOL
		# 1ra Eval. Termino 2007. Tema 1. Hormiga y arroz
		# Propuesta de solución. edelros@espol.edu.ec
		import random
		import math
		xh=int(input('coordenada x hormiga: '))
		yh=int(input('coordenada y hormiga: '))
		
		# Procedimiento
		xa=10
		
		ya=8
		
		dmayor=math.sqrt((xh-xa)**2+(yh-ya)**2)
		
		encontrado=0
		
		turno=0

En la solución con Python, se usarán las librerías básicas de aleatorios (random) y matemáticas (math).

Una alternativa a esta solución es usar la librería numérica NUMPY, que se descarga e instala como un módulo complementario.

Descripción	DIAGRAMA DE FLUJO	Python
Mientras los turnos no lleguen a 100 y no se encontró el arroz		`while (turno<100 and encontrado==0):`
Aleatorio para dirección		`dir=int(random.random()*4)+1`
Aleatorio para pasos		`pasos=int(random.random()*3)+1`
Movimiento arriba		`if dir==1:` `yh=yh+pasos`
Movimiento abajo		`if dir==2:` `yh=yh-pasos`
Movimiento derecha		`if dir==3:` `xh=xh+pasos`
Movimiento izquierda		`if dir==4:` `xh=xh-pasos`
Prueba hipótesis si se encontró el arroz		`if (xh==xa and yh==ya):` `encontrado=1`
Cálculo de distancia entre hormiga y arroz		`d=math.sqrt((xh-xa)**2+(yh-ya)**2)`
Prueba de hipótesis si la distancia resulto mayor que lo anteriormente conocido		`if d>dmayor:` `dmayor=d`
Cuenta turnos		`turno=turno+1`
Repita		

Muestra si lo encontró
Muestra los turnos
Muestra la distancia mayor
		`# Salida`
		`print('estado encontrado: ')`
		`print(encontrado)`
		`print('turnos simulados: ')`
		`print(turno)`
		`print('distancia más lejana: ')`
		`print(dmayor)`

Fin		

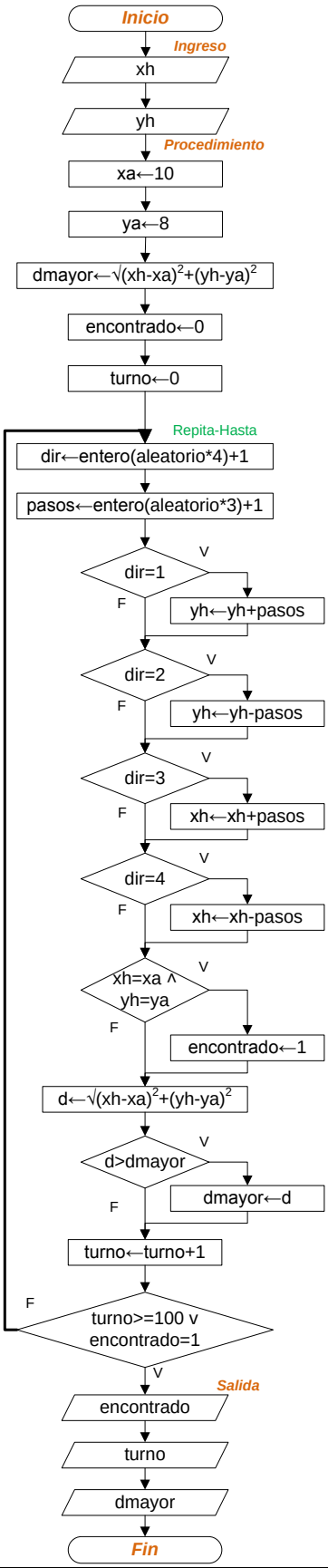
Ejecución del algoritmo: hormiga.py

```
>>>
coordenada x hormiga: -2
coordenada y hormiga: 2
estado encontrado:      1
turnos simulados:     70
distancia más lejana:     26.2488
```

```
>>>
coordenada x hormiga: -2
coordenada y hormiga: 2
estado encontrado:      0
turnos simulados:     100
distancia más lejana:     43.4626
```

Otra solución, usando lazo "Repita-Hasta", modificando el while con la negación de la expresión del diagrama, puesto que python no tiene soporte para este lazo.

Diagrama de Flujo



Python

```
# ICM00794-Fundamentos de Computación - FCNM-ESPOL
# 1ra Eval. Termino 2007. Tema 1. Hormiga y arroz
# Propuesta de solución. edelros@espol.edu.ec
import random
import math
xh=int(input('coordenada x hormiga: '))
yh=int(input('coordenada y hormiga: '))

# Procedimiento
xa=10
ya=8

dmayor=math.sqrt((xh-xa)**2+(yh-ya)**2)

encontrado=0

turno=0

while not(turno>=100 or encontrado!=0):
    dir=int(random.random()*4)+1

    pasos=int(random.random()*3)+1

    if dir==1:

        yh=yh+pasos


    if dir==2:

        yh=yh-pasos

    if dir==3:

        xh=xh+pasos

    if dir==4:

        xh=xh-pasos

    if (xh==xa and yh==ya):

        encontrado=1


    d=math.sqrt((xh-xa)**2+(yh-ya)**2)

    if d>dmayor:

        dmayor=d

    turno=turno+1



# Salida
print('estado encontrado: ')
print(encontrado)
print('turnos simulados: ')
print(turno)
print('distancia más lejana: ')
print(dmayor)
```