

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

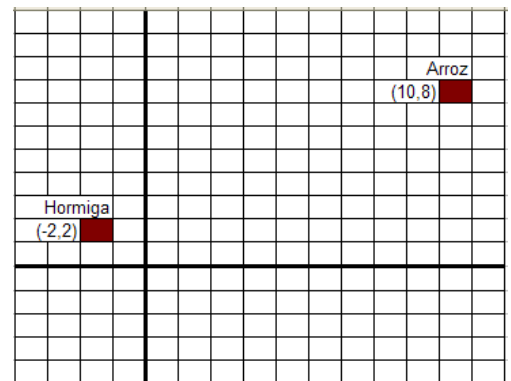
Tema 1 (30 puntos) En un plano cartesiano se encuentran una hormiga y un grano de arroz.

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:

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



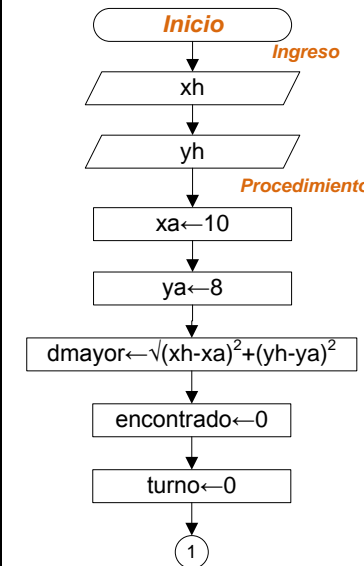
$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

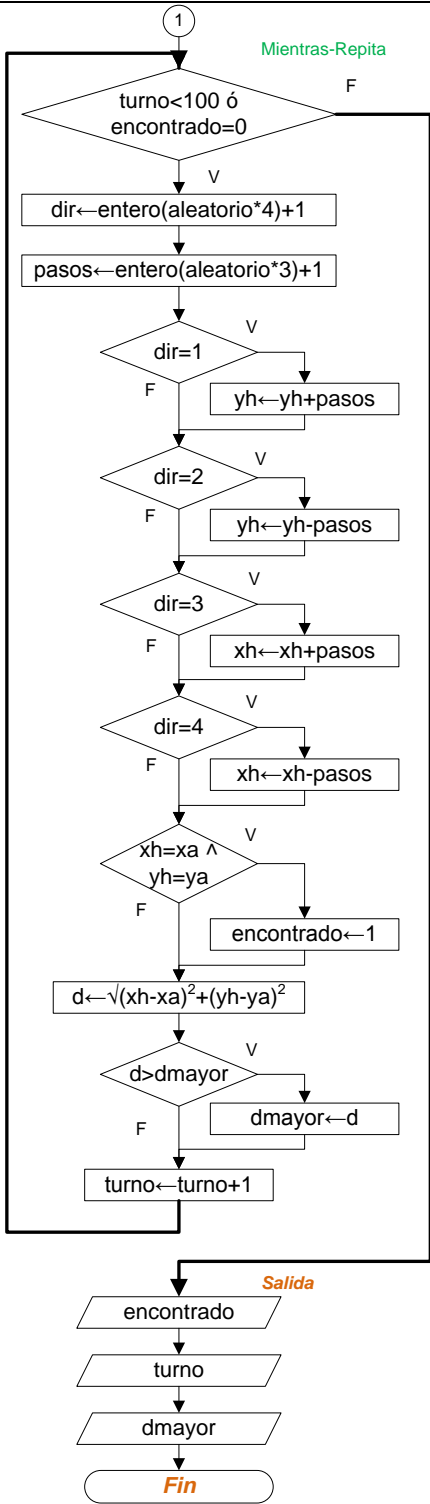
Rúbrica: Manejo de aleatorios (5 puntos), Control de movimiento y a) (10 puntos), b) (5 puntos) y c) (5 puntos). Solución integral (5 puntos)

Propuesta de Solución:

Considere ingresar la ubicación inicial **(xh,yh)** de la hormiga y que la posición del arroz **(xa,ya)** es fija. Suponemos que la distancia inicial es la distancia mayor **dmayor**, la hormiga no ha encontrado el grano de arroz. Para la **dirección** del movimiento de la hormiga y la cantidad de **pasos** se generas 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. Se cuenta un turno completado, y se repite todo hasta que se completen los 100 turnos o se haya encontrado el grano de arroz. Se termina mostrando los resultados buscados.

Tarea: Realizar las modificaciones para mostrar la cantidad de pasos en lugar de turnos.

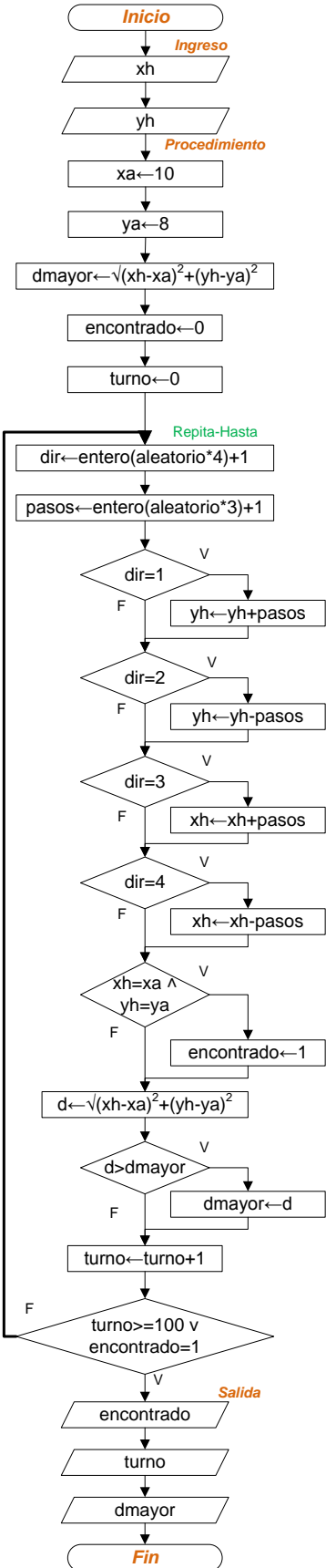
Descripción	DIAGRAMA DE FLUJO	OCTAVE/MATLAB
Inicio		<pre>% ICM00794-Fundamentos de Computación - FCNM-ESPOL % 1ra Evaluacion II Termino 2007. Tema 1.Hormiga y arroz % Propuesta de solución. edelros@espol.edu.ec xh=input('coordenada x hormiga: '); yh=input('coordenada y hormiga: '); % Procedimiento xa=10; ya=8; dmayor=sqrt((xh-xa)^2+(yh-ya)^2); encontrado=0; turno=0;</pre>
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		

Descripción	DIAGRAMA DE FLUJO	OCTAVE/MATLAB
<p>Mientras los turnos no lleguen a 100 y no se encontró el arroz</p> <p>Aleatorio para dirección</p> <p>Aleatorio para pasos</p> <p>Movimiento arriba</p> <p>Movimiento abajo</p> <p>Movimiento derecha</p> <p>Movimiento izquierda</p> <p>Prueba hipótesis si se encontró el arroz</p> <p>Cálculo de distancia entre hormiga y arroz</p> <p>Prueba de hipótesis si la distancia resulto mayor que lo anteriormente conocido</p> <p>Cuenta turnos</p> <p>Repita</p> <p>Muestra si lo encontró</p> <p>Muestra los turnos</p> <p>Muestra la distancia mayor</p> <p>Fin</p>	 <pre> graph TD Start((1)) --> Loop{turno < 100 ó encontrado == 0} Loop -- V --> Dir[dir ← entero(aleatorio*4)+1] Dir --> Pasos[pasos ← entero(aleatorio*3)+1] Pasos --> Dir1{dir=1} Dir1 -- V --> Yh[yh ← yh+pasos] Dir1 -- F --> Dir2{dir=2} Dir2 -- V --> Yh2[yh ← yh-pasos] Dir2 -- F --> Dir3{dir=3} Dir3 -- V --> Xh[xh ← xh+pasos] Dir3 -- F --> Dir4{dir=4} Dir4 -- V --> Xh2[xh ← xh-pasos] Dir4 -- F --> Found{xh=xa ^ yh=ya} Found -- V --> Found1[encontrado ← 1] Found -- F --> Dist[d ← √(xh-xa)² + (yh-ya)²] Dist --> DistCheck{d > dmayor} DistCheck -- V --> Dmayor[dmayor ← d] DistCheck -- F --> Turno[turno ← turno+1] Found1 --> Turno Turno --> Loop Loop -- F --> End([Fin]) Found1 --> End1[/encontrado/] Turno --> End2[/turno/] Dmayor --> End3[/dmayor/] </pre>	<pre> while (turno<100 && encontrado==0) dir=fix(rand*4)+1; pasos=fix(rand*3)+1; if dir==1 yh=yh+pasos; end if dir==2 yh=yh-pasos; end if dir==3 xh=xh+pasos; end if dir==4 xh=xh-pasos; end if xh==xa && yh==ya encontrado=1; end d=sqrt((xh-xa)^2+(yh-ya)^2); if d>dmayor dmayor=d; end turno=turno+1; end % Salida disp('estado encontrado: '); disp(encontrado); disp('turnos simulados: '); disp(turno); disp('distancia más lejana: '); disp(dmayor); </pre>

Ejecución del algoritmo: hormiga.m

<pre> >> hormiga coordenada x hormiga: -2 coordenada y hormiga: 2 estado encontrado: 1 turnos simulados: 70 distancia más lejana: 26.2488 </pre>	<pre> >> hormiga coordenada x hormiga: -2 coordenada y hormiga: 2 estado encontrado: 0 turnos simulados: 100 distancia más lejana: 43.4626 </pre>
--	---

Otra solución, usando lazo “Repita-Hasta”, ejecutado en Octave, puesto que matlab no tiene soporte para este lazo.

Diagrama de Flujo	OCTAVE
 <pre> graph TD Inicio([Inicio]) --> Ingreso[/Ingreso/] Ingreso --> xh[/xh/] xh --> yh[/yh/] yh --> Proc[Procedimiento] Proc --> xa[xa ← 10] Proc --> ya[ya ← 8] Proc --> dmajor["dmayor ← √(xh-xa)²+(yh-ya)²"] Proc --> encontrado0[encontrado ← 0] Proc --> turno0[turno ← 0] Proc --> RepitaHasta[Repita-Hasta] RepitaHasta --> dir["dir ← entero(aleatorio*4)+1"] RepitaHasta --> pasos["pasos ← entero(aleatorio*3)+1"] RepitaHasta --> Dir1{dir=1} Dir1 -- V --> yhp["yh ← yh+pasos"] Dir1 -- F --> Dir2{dir=2} Dir2 -- V --> yhm["yh ← yh-pasos"] Dir2 -- F --> Dir3{dir=3} Dir3 -- V --> xhp["xh ← xh+pasos"] Dir3 -- F --> Dir4{dir=4} Dir4 -- V --> xhm["xh ← xh-pasos"] Dir4 -- F --> Found{ "xh=xa ∧ yh=ya" } Found -- V --> encontrado1[encontrado ← 1] Found -- F --> d["d ← √(xh-xa)²+(yh-ya)²"] d --> DgtDmajor{d > dmayor} DgtDmajor -- V --> dmayor["dmayor ← d"] DgtDmajor -- F --> turno["turno ← turno+1"] turno --> LoopExit{ "turno ≥ 100 v encontrado=1" } LoopExit -- F --> RepitaHasta LoopExit -- V --> Salida[/Salida/] Salida --> encontrado[/encontrado/] Salida --> turno[/turno/] Salida --> dmajor[/dmayor/] Salida --> Fin([Fin]) </pre>	<pre> % ICM00794-Fundamentos de Computación - FCNM-ESPOL % 1ra Eval. II Terminio 2007. Tema 1. Hormiga y arroz % Propuesta de solución. edelros@espol.edu.ec. xh=input('coordenada x hormiga: '); yh=input('coordenada y hormiga: '); % Procedimiento xa=10; ya=8; dmayor=sqrt((xh-xa)^2+(yh-ya)^2); encontrado=0; turno=0; do dir=fix(rand*4)+1; pasos=fix(rand*3)+1; if dir==1 yh=yh+pasos; end if dir==2 yh=yh-pasos; end if dir==3 xh=xh+pasos; end if dir==4 xh=xh-pasos; end if xh==xa && yh==ya encontrado=1; end d=sqrt((xh-xa)^2+(yh-ya)^2); if d>dmayor dmayor=d; end turno=turno+1; until (turno>=100 encontrado==1) % Salida disp('estado encontrado: '); disp(encontrado); disp('turnos simulados: '); disp(turno); disp('distancia más lejana: '); disp(dmayor); </pre>