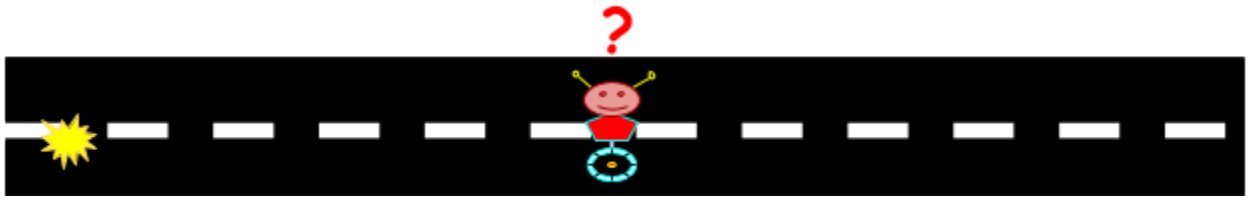
Robot Treasure Hunt

Minimize Total Distance



The intrepid little robot **ZBeepl/O** must find a new energy source or **ZB** is in **big** trouble. **ZB** cannot see or sense the energy in any way until directly touching it. **ZB** may only move left or right along the road. **ZB** must also move the **least** possible distance, conserving energy, not knowing how far it is to the new energy. If **ZB** fails to find new energy, the evil Bart Tater gets **all** the cookies. There will only be **one** new energy source found on the road. The new energy source is guaranteed accessible, just barely, if **ZB** uses the correct strategy to find it. Maybe the new energy is **10,000m** distant.

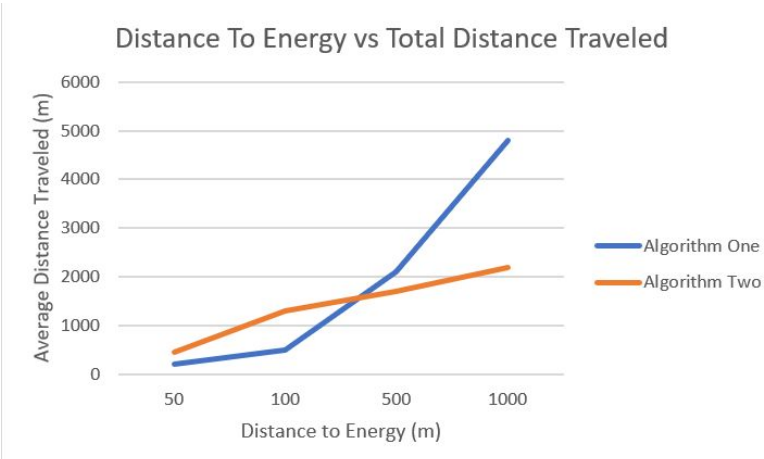
Determine **two** different ways **ZB** might move left/right to find the energy. Create a table showing the total distance traveled using the distances shown below. Use meters as the unit for distance. Remember, **minimizing** the total distance traveled is the goal. Describe below the two algorithms you use. Does it matter which direction **ZB** decides to go first?

Algorithm One: Move 5m distance out and back, switching direction, add 5m to travel distance each time **ZB** crosses the starting point.

Algorithm Two: Randomly choose one direction and go all the way until reaching the wall, if the energy is not found by then, turn around and go all the way in the opposite direction until hitting the energy.

Check Out The Greenfoot code I used for each algorithm below!

Distance to Treasure		Average Total Distance Traveled
50m		One: 200m Two: 450m
100m		One: 490m Two: 1300m
500m		One: 2105m Two: 1700m
1000m		One: 4810m



		Two: 2200m	

Distance to Treasure	Algorithm One	Algorithm Two
50m	Attempt 1: 290m Attempt 2: 290m Attempt 3: 140m Attempt 4: 140m Attempt 5: 140m Average: 200m	Attempt 1: 50m Attempt 2: 2050m Attempt 3: 50m Attempt 4: 50m Attempt 5: 50m Average: 450m
100m	Attempt 1: 590m Attempt 2: 340m Attempt 3: 590m Attempt 4: 340m Attempt 5: 590m Average: 490m	Attempt 1: 2100m Attempt 2: 100m Attempt 3: 2100m Attempt 4: 2100m Attempt 5: 100m Average: 1300m
500m	Attempt 1: 1490m Attempt 2: 2515m Attempt 3: 1490m Attempt 4: 1490m Attempt 5: 2515m Average: 2105m	Attempt 1: 500m Attempt 2: 2500m Attempt 3: 500m Attempt 4: 2500m Attempt 5: 2500m Average: 1700m
1000m	Attempt 1: 5240m Attempt 2: 5240m Attempt 3: 3090m Attempt 4: 5240m Attempt 5: 5240m Average: 4810m	Attempt 1: 1000m Attempt 2: 3000m Attempt 3: 3000m Attempt 4: 3000m Attempt 5: 1000m Average: 2200m

Algorithm One Code

```
/**
 * Constructor for objects of class MyWorld.
 *
 */
public MyWorld()
{
    // Create a new world with 600x400 cells with a cell
    super(3000, 350, 1);
    setBackground("desert.jpg");
    drawRoad();
    prepare();
}
```

```
public void drawRoad(){
    GreenfootImage bg = getBackground();

    GreenfootImage road = new GreenfootImage(3000,40);
    road.setColor(Color.BLACK);
    road.fillRect(0,0,3000,40);

    GreenfootImage stripe = new GreenfootImage(10,3 );
    stripe.setColor(Color.YELLOW);
    stripe.fillRect(0,0,10,3);
    // Draw Stripes
    for(int i = 3; i < 3000; i+=20){
        road.drawImage(stripe, i, 17);
    }

    // Draw stert line
    GreenfootImage startLine = new GreenfootImage(5,40);
    startLine.setColor(Color.RED);
    startLine.fillRect(0,0,5,40);
    road.drawImage(startLine, 1496, 0);

    // Draw Road
    bg.drawImage(road, 0, 240);
}
```

```
private int direction = 1;
private int currentGoal = 5;
private int progress = 0;
private int moveAmount = 5;
private int total = 0;
private int counter = 0;
/**
 * Act - do whatever the Robot wants to do
 * the 'Act' or 'Run' button gets pressed
 */
public void act()
{
    move(moveAmount * direction);
    counter += 1;
    total += 5;
    progress += 1;
    if(progress == currentGoal){
        progress = 0;
        currentGoal += currentGoal;
        turn(180);
        GreenfootImage img = getImage();
        img.mirrorVertically();
        setImage(img);
    }
    if(isTouching(Energy.class)){
        Greenfoot.stop();
    }
}
```

```
private int x = location();
/**
 * Act - do whatever the Energy wants to do. T
 * the 'Act' or 'Run' button gets pressed in t
 */
public void act()
{
    setLocation(x, 240);
}
public int location(){
    if(Greenfoot.getRandomNumber(101) > 50){
        return (1540 + 1000);
    }
    else{
        return (1440 - 1000);
    }
}
```

Algorithm Two Code

```
public MyWorld()
{
    // Create a new world with 600x400 cells with a cell
    super(2000, 350, 1);
    setBackground("desert.jpg");
    drawRoad();
    prepare();
}
```

```
public void drawRoad(){
    GreenfootImage bg = getBackground();

    GreenfootImage road = new GreenfootImage(2000,40);
    road.setColor(Color.BLACK);
    road.fillRect(0,0,2000,40);

    GreenfootImage stripe = new GreenfootImage(10,3 );
    stripe.setColor(Color.YELLOW);
    stripe.fillRect(0,0,10,3);
    // Draw Stripes
    for(int i = 3; i < 2000; i+=20){
        road.drawImage(stripe, i, 17);
    }

    // Draw stert line
    GreenfootImage startLine = new GreenfootImage(5,40);
    startLine.setColor(Color.RED);
    startLine.fillRect(0,0,5,40);
    road.drawImage(startLine, 1000, 0);

    // Draw Road
    bg.drawImage(road, 0, 240);
}
```

```
private int counter = 0;
private int x = Greenfoot.getRandomNumber(101);
private int z = setDirection(x);
public void act()
{
    move(1);
    counter += 1;
    setRotation(z);
    checkEdge();
    if(isTouching(Energy.class)){
        Greenfoot.stop();
    }
}
public int setDirection(int y){
    if(y > 50){
        return 0;
    }
    else{
        GreenfootImage img = getImage();
        img.mirrorVertically();
        setImage(img);
        return 180;
    }
}
public void checkEdge(){
    if(getRotation() == 0 && isAtEdge()){
        z = 180;
        move(3000);
        GreenfootImage img = getImage();
        img.mirrorVertically();
        setImage(img);
        counter += 1050;
    }
    if(getRotation() == 180 && isAtEdge()){
        z = 0;
        move(3000);
        GreenfootImage img = getImage();
        img.mirrorVertically();
        setImage(img);
        counter += 1050;
```

```
private int x = location();
/**
 * Act - do whatever the Energy wants to do.
 * the 'Act' or 'Run' button gets pressed in
 */
public void act()
{
    setLocation(x, 240);
}
public int location(){
    if(Greenfoot.getRandomNumber(101) > 50){
        return (1009 + 1050);
    }
    else{
        return (994 - 1050);
    }
}
```