

Name: _____ Class: _____

At the SLAM station in the fulfillment center, Amazon checks each customer order to ensure that the right contents are included in their shipment. Each box is weighed as it travels down a conveyor belt. If the box weighs the “right” amount, the box continues forward and will have an address label applied. If the box does not weigh the right amount, it will be pulled off and checked by an Amazon associate. The SLAM station completes this quality control process to check thousands of boxes each hour. Can you dissect this algorithm?

Directions: Answer the guiding questions below to help you dissect the SLAM algorithm. Then, use a flowchart or pseudocode to write your own SLAM algorithm. You will present your algorithm to the class for feedback and then have a chance to refine using ideas from others.

1. How does Amazon know how much each box should weigh? How would they calculate that?
2. What data will your algorithm need to call? List out all the data your algorithm will need, where it will be called from, and a variable name for each.

Data needed:	Where will this data come from?	Variable Name:

3. Use pseudocode or a flowchart to write out your algorithm for the SLAM station.
4. Algorithms aren't fool-proof. What could go wrong even with this quality control system?
5. SLAM processes thousands of boxes every hour. What does this show us about computer science and cloud computing power?

Thank you for joining us on an Amazon Future Engineer Tech Tour!
Keep exploring at AmazonFutureEngineer.com.



Example Answers

Below are a few examples of answers you might see from students. Use these as a launching point to provide feedback to your students.

1. How does Amazon know how much each box should weigh? How would they calculate that?

Amazon calculates the weight that the box should be by referencing the items' weight from their database and adding those to see if it matches the current box's weight. They call data from the database and use a sensor to weigh the box.

2. What data will your algorithm need to call? List out all the data your algorithm will need, where it will be called from, and a variable name for each.

Data needed:	Where will this data come from?	Variable Name:
Expected Item Weights	Database—the algorithm will need to reference each item's weight in the order and add it together to create the full item weight.	expectedItemWeight1 expectedItemWeight2 expectedItemWeight3
Actual Weight	Sensor—the conveyor belt sensor will weigh the box to calculate its total weight.	actualWeight
Order Number	Label on Box—camera sensor scans it to call order information.	orderNumber

3. Use pseudocode or a flowchart to write out your algorithm for the SLAM station.

//natural language example

Use sensor to read rawWeight.

If actualWeight <= 0, then turn light red and PAUSE.

Scan box label and update orderNumber.

Use orderNumber to identify expectedItemWeights of items in order and expectedBoxWeight.

Add expectedItemsWeights and expectedBoxWeight and assign to expectedTotalWeight.

If expectedTotalWeight - actualWeight > 0.001 pounds or < -0.001 pounds, then turn light red and PAUSE. Turn light green and continue.

// More code-like example

actualWeight <- weight sensor reading

If actualWeight <= 0: Turn light red and PAUSE.

orderNumber <- scan box label

expectedItemWeights <- get list of expected item weights from database for items in orderNumber order expectedBoxWeight <-

get expected box weight from database for box in orderNumber order

expectedTotalWeight <- sum(expectedItemsWeights) + expectedBoxWeight

If expectedTotalWeight - actualWeight < -0.001 pounds or > 0.001 pounds: // check weight within tolerance [NOTE: could also use a constant MAX_WEIGHT_TOLERANCE = 0.001]

Turn light red and PAUSE

Turn light green and continue

