APM 153 LECTURE TWO – Basics of Algorithm Design, Making Decisions in an Algorithm, Flowcharts, Getting Started Using Matlab

Basics of Algorithm Design

(1) Designing an algorithm is easy if you follow a few simple steps. First, **state the problem**. Write a short sentence or two that states what it is you want to do.

(2) For example, suppose you were given seven coins. Six of the coins weigh exactly the same, but the seventh coin is slightly heavier. Using only a balance scale (a scale which balances when equal weights are placed on both sides), how can you determine which one is the heavy coin?

State the Problem _____

_____

_____

(3) The next step is to **define the inputs and outputs**. Usually when we talk about inputs and outputs we are talking about **data.** In this case, the "inputs" are the coins we put onto the balance scale and the "outputs" are the results (that is whether the two sides balance or not).

(4) Last, list the series of steps in the algorithm needed to solve the problem. Some of these steps may be **repeated**. The entire series of steps in this case will include

A. **input** (putting some coins on one side or the other of the balance)
B. **output** (seeing whether the scale balances)
C. **a decision** (which coin is the heavy one)

(5) The trick to this problem is deciding how many coins to put on each side of the balance. The other trick is to see how few times you need to use the balance before you find the one heavy coin. Write out your algorithm on the lines below.

A. Put ____ coin(s) on _____side(s) of the balance.

B. _____

C. _____

_____

(6) How would you change the algorithm if you had eight coins with one heavy coin?

A. Put ____ coin(s) on _____side(s) of the balance.

B. _____

C. _____

_____

Making Decisions in an Algorithm

(7) In both of the examples above, whether you had 7 coins or 8 coins, you have to make some decisions within the algorithm. If you find the heavy coin on the first trial, you are done. If you don't find the heavy coin on the first trial you have to make another trial until you find the heavy coin.

(8) In both cases, the decision you have to make takes the form of an "If – Then" statement. For example, "**If** the two sides of the pan balance, **then** the heavy coin is the seventh coin still sitting on the table."

(9) If-Then statements are an important part of writing algorithms. In a computer program based upon our coin-weighing algorithm, we might have line of code in the program that tells the computer,….

*"**If** the two sides balance, **then** the heavy coin is the seventh coin. **STOP**"*

followed by another line of code that tells the computer,…

*"**If** the two sides of the pan don't balance, **then** put one of the coins from the heavy side of the balance on one pan, put a second coin from the heavy side on the other pan, and leave the third coin on the table."*

In turn, this line of code would be followed by another line that told the computer,…

*"**If** the two sides of the pan balance, **then** the coin on the table is the heavy coin. **STOP**"*

And last, there would be a line of code that said,…

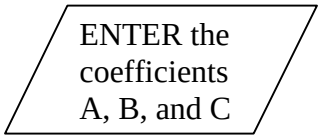
*"**If** the two side do not balance, **then** heavy coin is the one on the heavy side. **STOP**."*

(10) In every case in the imaginary program above, when you have finally found the heavy coin, the program is told to "stop" or END.

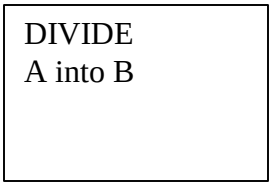
Flowcharts

(11) There is a very simple, formal set of symbols which can help us create algorithms and design programs. Most algorithms and programs can be represented using just the following four flowchart symbols.

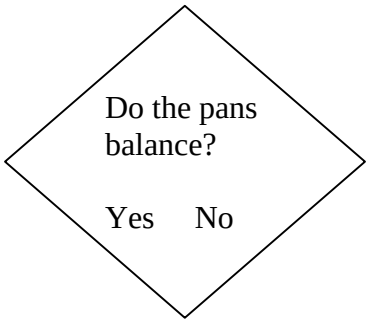
"**Terminator**" - This symbol represents the START and the END of an algorithm or program. Algorithms and programs can have more than one END where the algorithm or program terminates.



"**Data**" - This symbol is used to represent INPUT and OUTPUT. If the algorithm calls for data to be entered, we use this symbol. Where the program will show us the result, we use this symbol as well. Often, we will write inside the box a note telling us what data we mean.



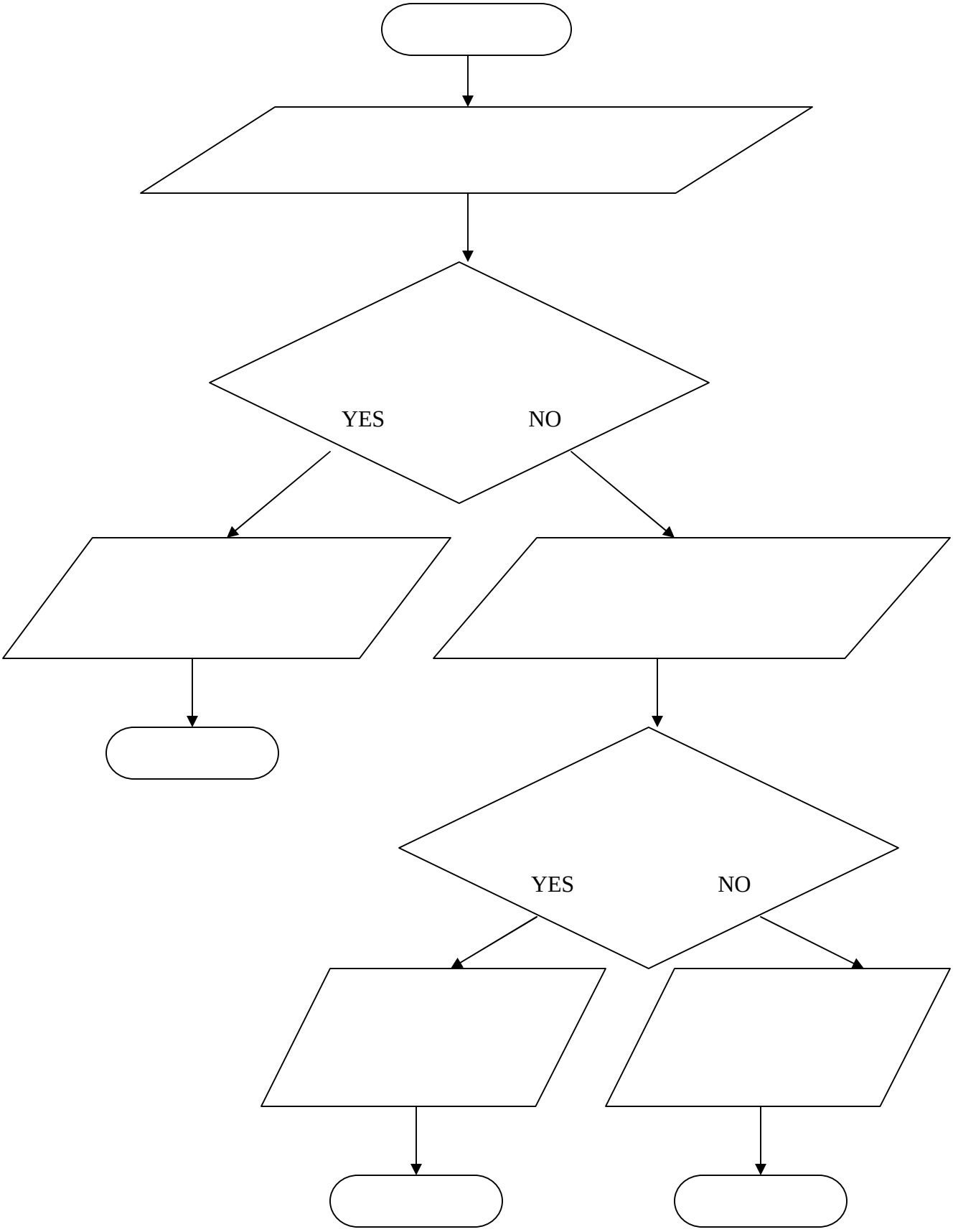
"**Process**" – This is the symbol that represents a CALCULATION. For example if we wanted to divide one number by another.



"**Decision**" – This is the symbol used to represent an IF-THEN statement. Inside the diamond we write the If-Then statement and depending upon the answer, the algorithm will "branch" off from one side of the box or the other.

(12) We can see what is meant by "branching" when we connect all the symbols together.

(13) Take a look at the flowchart below and fill in the symbols with the words for our coin weighing algorithm.


YES
NO
YES
NO

(14) You can access the flowchart and arrow symbols in Microsoft Word by clicking on the word INSERT in the toolbar, choosing PICTURE, and then choosing AUTOSHAPES.

(15) Double-clicking on the icons representing the flowchart and arrow symbols brings up menus for these symbols that will remain on the MS Word desktop until you close them.

Getting Started Using Matlab

(16) The reading assignment you were given on the first day of class was to read chapters one and two in your textbook *Matlab Programming for Engineers*.

(17) While you are reading the first two chapters, you may want to actually open the Matlab program in one of the computer clusters and type in the commands shown in the book.

(18) Whether you run Matlab while you are doing your reading assignment or not, after you have done the reading assignment, you should open up Matlab and run through some of the DEMOS.

(19) When Matlab starts, you are shown what is known as a **graphic user interface** or GUI for short. The GUI consists of four windows which are described in the reading and in the demos.

(20) When Matlab starts, you will see the word DEMOS in the command window on the right hand side of the screen. Click once on the word DEMOS.

(21) I recommend running through the *What is Matlab* and the *Matrices and Arrays* demos.

.