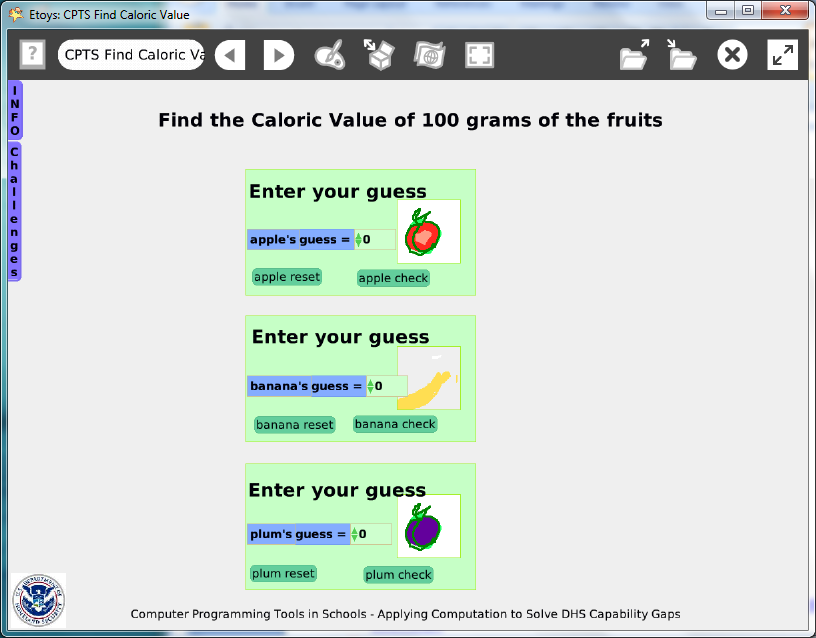
**Project: Caloric Value**

**Difficulty: Level 2**

****

Project: <http://etoysillinois.org/library?tags=Computer%20Programming%20Tools%20in%20Schools>

Video Preview: <http://www.youtube.com/watch?v=kvLDueiWJEM>

The Preview Video's lesson worksheets focus on observations and analysis of events in the video; see page 10.

Time: Four 45 minute labs

**Challenge:** Design a project game that uses facts about healthy food’s caloric values and allows the user to interact, to guess and to guess again until they find correct answers.

**Programming:**

This project uses conditional statements, variables, constructed tiles. and buttons to fire scripts. This project is not difficult programming and once the scripts are finished for one fruit the remaining fruits in the games are copies that are repainted, renamed, and whose caloric values are changed. Once students realize how easy it is to add more fruits to the game, theirs may become much more complicated than this sample project. Attention to detail when building the first object and its scripts is repaid by the benefit of reusing the scripts over and over again. This project uses text as a set of characters for the announcement and to report whether the guess was too high or too low. There are no Quick Guides about text as characters so give students time to experiment with this category of tiles.

**Things you’ll need to know:**

Quick Guides

* Paint Tools/ All
* Halo Handles/All
* Supplies: Text, Add a New Flap
* Script Tiles: Tests Category
* Menus: Normal Ticking, Viewer Icons Set, Scriptor Icons Set, Button Fires a Script, Watchers

Note that each fruit, its scripts and buttons are on a playfield. This keeps things organized and a playfields can be collapsed when working space is needed for other objects.

**Things to think about:**

Text is an object and its characters can be used as label but can change to show whether a guess is too high or too low.

**Extensions:**

Ask students to compare the use of characters within a text to the separate texts where what showed and what was hidden was controlled in a conditional statement. Which is easier, which is more powerful?

**NETS for Students:**

<http://www.iste.org/standards/nets-for-students/nets-student-standards-2007.aspx>

1. Creativity and Innovation: a, b

2. Communication and Collaboration: b

3. Research and Information Fluency: a, b

4. Critical Thinking, Problem Solving, and Decision Making: a, b

5. Digital Citizenship: a, b

6. Technology Operations and Concepts: a, b, d

**CSTA:**

CSTA Level II: Objectives and Outline

<http://csta.acm.org/Curriculum/sub/CurrFiles/L2-Objectives-and-Outlines.pdf>

Level II objectives for middle school students are furthered through studying a programming language well enough that the student is proficient with it. Whether the language is Etoys, StarLogo TNG, or Scratch, it is the ability to use the language to express ideas that is valuable. A student skillful enough to use *any* programming language to express ideas, solve problems, model behaviors, simulate data, or to educate or entertain is an entitled person in today’s society. Topics of particular note are:

Topic 2: Problem Solving

Topic 6: Connections between Mathematics and Computer Science

Topic 11: Programming Languages

Topic 13: Multimedia

**Common Core Standards Mathematics:**

<http://www.corestandards.org/the-standards/mathematics>

6. RP.3

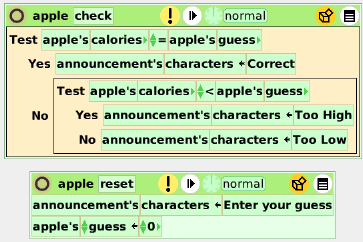
8. SP.4

**Materials:**

Ask students to read current news articles about serving sizes and calories. Some fast food restaurants are posting the calorie count for menu items and some cities are passing laws limiting serving sizes.

**Comments: Objects - Scripts – Decisions**

Example Scripts:



**Teacher Notes:**

Caloric Value

This Etoys applet allows students to use different guessing techniques to determine the exact number of calories in 100 grams of a fruit. The students will learn a bit about the different caloric values of fruits while also investigating several different algorithms for homing in on an exact value. These “searching algorithms” are an important idea in computer science.

Here are a few suggestions about how this applet might be used in the classroom.

1) In its most simple form, students can just play with the applet; guessing without any guidance. This may become tiresome for them, as the caloric value must be exact to get a “Correct”. It is hoped they would begin to formulate “methods” for guessing. After the correct calorie count has been determined for each fruit, the class can discuss why they differ, exactly what 100 grams of the fruit may look like or what they think an “average serving size” of each fruit might be.

2) If you wish to use the applet to teach “guessing” algorithms, then you can use the following “My Guess Is Better Than Yours” worksheet to have the students practice the different methods. The three common searching algorithms covered by the worksheet are called “linear” and “binary” and “hashing”. Have them go back to the applet and discuss using each method.

3) If the students know Etoys programming, you could ask them to research different fruits and add them to the applet. They would need to program them to produce the correct responses, and perhaps expand the game so that the new fruits would show up when the original three were done.

**My Guess is Better Than Yours** Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The work of finding the exact numeric answer to a question is often left to a computer. We all know a computer can’t “think”, so how does it find the answer? Below are two methods programmers often use.

1) **Linear Search:** Linear in this sense means “in a line”. Since numbers are generally sorted by size (*in counting it is smallest to largest*), we can have the computer start with the smallest and just check each number in the line until it finds the correct answer. When a number is reached the computer asks, “Is this number the correct answer?” If it gets “yes” then it is done. If it gets the answer “No”, then it moves to the next larger number and asks the question again.

a) If the number chosen is 8, how many questions would the computer have to ask to find the correct answer of 8? \_\_\_\_\_\_\_ What if the number chosen was 14? \_\_\_\_\_\_\_

b) When would this seem like an efficient way to search? When does it seem less efficient?

c) Why do you think this method is used for computers?

2) **Binary Search:** Binary here means “cut into two”. The first thing the computer does is cut the largest number in half. This means it now has two smaller sets of numbers and then asks, “Is this number the correct answer?” If the answer is “no”, then it asks “Is it smaller than the correct number?” If the answer is “no” then it knows the correct answer is in the lower half of the original set. It then takes the upper half, cuts it in half and repeats the process until it gets the correct answer.

1. Let’s look at the question from above: Chose an integer from 1 to 20.

Let’s chose 8.

Cut the set into two smaller sets: 1-10 and 11-20.

1 2 3 4 5 6 7 **8**  9 10 11 12 13 14 15 16 17 18 19 20

Is 10 correct? \_\_\_\_\_\_\_ Is 10 too small? \_\_\_\_\_\_\_\_

b) Since 10 is not too small, the correct answer is between 1 and 10. I cut ten into two groups:

1 2 3 4 5 6 7 **8** 9 10

Is 5 correct? \_\_\_\_\_\_\_\_ Is 5 too small? \_\_\_\_\_\_\_ Now what do I know?

c) What should I do next? Draw the sets below. What number do you think I will try first?

d) How many divisions will it take to find the correct answer? \_\_\_\_\_\_\_\_\_\_\_\_

e) How many divisions would it take if the correct answer was 14? \_\_\_\_\_\_\_\_\_\_\_\_

f) Why do you think this method is used for computing searches?

3) **Hash Search:** Hashing here means breaking a very large set into smaller pieces. It is a combination of the above two search algorithms. What you would do is first assume a “reasonable” search field. In the case of fruit, it would be reasonable to think 100 grams would never be more than 500 calories.

Then you decide how you want to “hash” it. Let’s say into hundreds. You would get 5 sets

1-100 101-200 201-300 301-400 401-500

You would then choose either a linear or a binary search of the five sets. When you found the set that contains the correct answer, you would then do either a linear or binary search of that set. In our case let’s say it is from 101 to 200. You could even do another “hash” on that set and break it into 10’s before you start checking for individual numbers.

a) Try finding 176 by binary and hashing by 10. Which do you think is best? Why?

Binary Hash by 10

101-150 151-200 100 110 120 130 140 150 160 170 180 190

**My Guess is Better Than Yours**

Answer Key

The work of finding the exact numeric answer to a question is often left to a computer. We all know a computer can’t “think”, so how does it find the answer? Below are two methods programmers often use.

1) **Linear Search:** Linear in this sense means “in a line”. Since numbers are generally sorted by size (*in counting it is smallest to largest*), we can have the computer start with the smallest and just check each number in the line until it finds the correct answer. When a number is reached the computer asks, “Is this number the correct answer?” If it gets “yes” then it is done. If it gets the answer “No”, then it moves to the next larger number and asks the question again.

a) If the number chosen is 8, how many questions would the computer have to ask to find the correct answer of 8? \_\_ 8\_\_ What if the number chosen was 14? \_\_­14\_\_\_\_

b) When would this seem like an efficient way to search? When does it seem less efficient?

*If there weren’t a lot of numbers to guess; if there were 10,000 numbers even a fast computer would be wasting time.*

c) Why do you think this method is used for computers?

*It seems very easy to program, and a computer can’t think. If the number of choices was small enough it could be done very quickly.*

2) **Binary Search:** Binary here means “cut into two”. The first thing the computer does is cut the largest number in half. This means it now has two smaller sets of numbers. Then asks:“Is this number the correct answer?” If the answer is “no”, then it asks “Is it smaller than the correct number?” If the answer is “no” then it knows the correct answer is in the lower half of the original set. It then takes the upper half, cuts it in half and repeats the process until it gets the correct answer.

a) Let’s look at the question from above: Chose an integer from 1 to 20. Let’s chose 8.

Cut the set into two smaller sets: 1-10 and 11-20.

1 2 3 4 5 6 7 **8**  9 10 11 12 13 14 15 16 17 18 19 20

Is 10 correct? \_\_\_No\_\_\_\_ Is 10 too small? \_\_\_No\_\_\_\_\_

b) Since 10 is not too small, the correct answer is between 1 and 10. I cut ten into two groups:

1 2 3 4 5 6 7 **8** 9 10

Is 5 correct? \_\_\_\_No\_\_\_\_ Is 5 too small? \_\_\_Yes\_\_\_ Now what do I know?

*Use the upper half.*

c) What should I do next? Draw the sets below. What number do you think I will try first?

*Split 6-10 into two sets (actually 3 sets)*

6 7 **8** 9 10

*Try the middle number first.*

*8 is the correct answer. DONE!*

d) How many divisions will it take to find the correct answer?

*Only 3 guesses!*

e) How many divisions would it take if the correct answer was 14?

*See a) above. Then it would need to be 11 -20 split into*

11 12 13 **14** 15 16 17 18 19 20

*Then it would be 11-15 split into three* 11 12 13 14 15.

*Checking 13 would result in having to check 14 and 15.*

*You would find 14 on your next guess.*

*This takes 4 guesses.*

f) Why do you think this method is used for computing searches?

*It takes fewer guesses, especially if there are lots of numbers! Every time the guesses get cut in half.*

3) **Hash Search:** Hashing here means breaking a very large set into smaller pieces. It is a combination of the above two search algorithms. What you would do is first assume a “reasonable” search field. In the case of fruit, it would be reasonable to think 100 grams would never be more than 500 calories.

Then you decide how you want to “hash” it. Let’s say into hundreds. You would get 5 sets

1-100 101-200 201-300 301-400 401-500

You would then choose either a linear or a binary search of the five sets. When you found the set that contains the correct answer, you would then do either a linear or binary search of that set. In our case let’s say it is from 101 to 200. You could even do another “hash” on that set and break it into 10’s before you start checking for individual numbers.

a) Using 101 to 200, try finding 176 by binary and by hashing by 10.

Which do you think is best? Why?

Binary Hash by 10

101-150 151-200 100 110 120 130 140 150 160 170 180 190

Checking gives 151-200 which becomes Linear search gives the 170 Hash with 7 questions,

151-175 176 -200 Binary searches get to the 170 Hash in 2 questions.

Checking gives 176-200 or Now you can use Linear or Binary to search this hash.

176-187 188 189-200 Linear searches would take 7 questions, Binary takes 3.

Checking gives 176-187 or Checking 170-179 or 170-174 175-179

176-181 182-187 gives 175-179

Checking gives 176-181 or Checking 175-179 or 175-176 177 178-179

176-178 179-181 gives 175-176

Checking gives 176-178 or Which gives 176.

176 177 178

Which gives 176

**Just Binary takes 6 guesses Hashing with Linear takes 14 questions**

**Hashing with Binary takes 5 questions**

**Caloric Value.mp4**

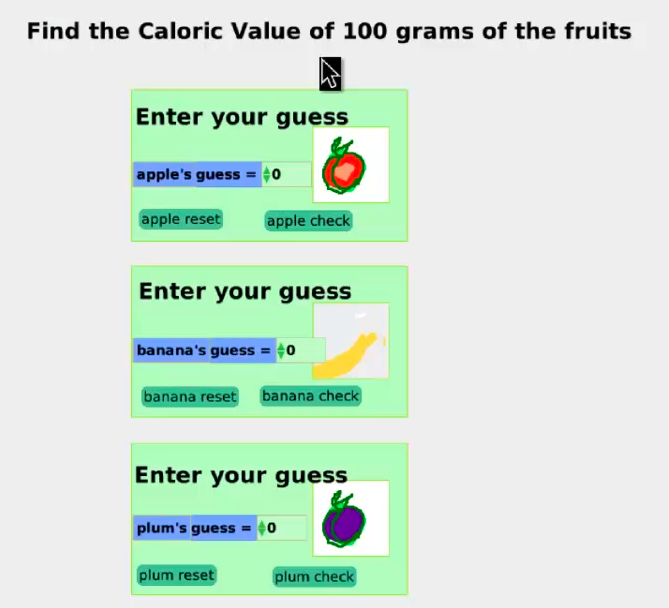
**Teacher Notes**

This little movie is designed to be used in three ways:

• You can show it to teachers, parents, administrators or other community members as a way to build interest in bringing computer science into the k-12 curriculum.

• You can show it to students to foster discussion about different guessing techniques to determine the exact number of calories in 100 grams of a fruit. (See Activity #1)

• You can show it as a model for a project you will later have the students build for themselves in Etoys. (See Activity #2)



**Caloric Value.mp4**

**Activity #1**

This Etoys applet allows students to use different guessing techniques to determine the exact number of calories in 100 grams of a fruit. The students will learn a bit about the different caloric values of fruits while also investigating several different algorithms for homing in on an exact value. These “searching algorithms” are an important idea in computer science. The purpose of this activity is to have the students watch the video clip of the applet and discuss what techniques they might suggest to most efficiently guess the exact number of calories.

The teacher should put the image of the movie up on the screen and before showing it ask the students to guess the correct number of calories for each of the three fruits. The movie should then be shown once to see who was closest to the correct number of calories for the apple. The students should then be put in groups of two or three told watch the movie in order to determine what they see as the “behaviors” of the guesser. They should fill in the worksheet (Guessing Calories) as they watch. They should be reminded that they may pause and backup as often as they like. The groups should also suggest alternative guessing strategies.

All groups should report out their observations, and then as a class discuss the findings. In Activity #2, three of the most common “search algorithms” used in computer science are discussed. It is hoped that some of these algorithms will come up in the conversation about the movie.

The lesson can be concluded by having the students research what 100 grams of an apple, banana, etc. looks like. Discuss why we usually refer to a “serving size” as a full fruit rather than an actual weight.

Guessing Calories Names\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1) What does 100 grams look like? Is an apple more or less than 100 grams?

2) List the values, in order, that the guesser used to arrive at the exact value.

3) Explain the strategy (method) you think the guesser was using to arrive at the answer.

4) If the guesser used the same strategy to find the banana calories as they used for the apple, what value do you think they would guess next and why? (Hint 100 calories is too high.)

5) The correct guess for the banana is 88 calories for 100 grams. Using the strategy from above, list the guesses you would use to arrive at the correct answer. Do you think this is an efficient strategy? Why or why not?

6) Explain another strategy you might use to guess the calories for the banana.

Guessing Calories

Answer Key

1) What does 100 grams look like? Is an apple more or less than 100 grams?

***100******grams*** *is equal to 3.5 ounces. For purposes of dieting,* ***100******grams*** *can equal*

*half an apple, two fried eggs, 1/2 cup of rice or 1/2 cup of cooked ground beef.*

<http://www.ask.com/answers/196480721/how-much-is-100-grams>

*The average apple is 200 grams or 108 calories.*

2) List the values, in order, that the guesser used to arrive at the exact value.

*50, 60, 55, 54*

3) Explain the strategy (method) you think the guesser was using to arrive at the answer.

*Went by 10’s and then by 5’s and then by 1’s.*

4) If the guesser used the same strategy to find the banana calories as they used for the apple, what value do you think they would guess next and why? (Hint 100 calories is too high.)

*They would go down 10 to 90*

5) The correct guess for the banana is 88 calories for 100 grams. Using the strategy from above, list the guesses you would use to arrive at the correct answer. Do you think this is an efficient strategy?

Why or why not?

*90, 80, 85, 86, 87, 88*

*They went down by 10 and still too high, down by another 10 and then too low so up by*

*still too low, and then up by ones*

*Not very efficient because they had to guess every number from 85 to 88*

6) Explain another strategy you might use to guess the calories for the banana.

*Answers will vary, but it is hoped that they might think of “halving” the space between the last two guesses each time. This is one of the standard algorithms used in computer “guessing”.*

*See Activity #2.*

**Caloric Value.mp4**

**Activity #2**

If you wish to use the applet to teach “guessing” algorithms, then you can use the following “My Guess Is Better Than Yours” worksheet to have the students practice the different methods. The three common searching algorithms covered by the worksheet are called “linear” and “binary” and “hashing”.

After they finish the worksheet they should download the Etoys applet at:

<http://etoysillinois.org/library?tags=Computer%20Programming%20Tools%20in%20Schools>

Click on “Caloric Value” and then “Download this Etoy”

They should run the applet several times using the three different methods from the worksheet to see how many guesses each method takes. The first time through they should start each guess with 100 calories.

The results will be: Search Guesses

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fruit | Linear | Binary | Hash | Best Search |
| Apple | 46 | 7 or 8 | 5 | Hash |
| Banana | 12 | 3 or 4 | 7 or 8 | Binary |
| Plum | 53 | 7 or 8 | 7 | Hash |

Each group should then try another starting estimate (or the teacher may give each group an estimate to start with) and they should see which method is best. In almost all cases, the linear method is the least efficient.

**My Guess is Better Than Yours** Name \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

The work of finding the exact numeric answer to a question is often left to a computer. We all know a computer can’t “think”, so how does it find the answer? Below are two methods programmers often use.

1) **Linear Search:** Linear in this sense means “in a line”. Since numbers are generally sorted by size (*in counting it is smallest to largest*), we can have the computer start with the smallest and just check each number in the line until it finds the correct answer. When a number is reached the computer asks, “Is this number the correct answer?” If it gets “yes” then it is done. If it gets the answer “No”, then it moves to the next larger number and asks the question again.

a) If the number chosen is 8, how many questions would the computer have to ask to find the correct answer of 8? \_\_\_\_\_\_\_ What if the number chosen was 14? \_\_\_\_\_\_\_

b) When would this seem like an efficient way to search? When does it seem less efficient?

c) Why do you think this method is used for computers?

2) **Binary Search:** Binary here means “cut into two”. The first thing the computer does is cut the largest number in half. This means it now has two smaller sets of numbers. Then asks: “Is this number the correct answer?” If the answer is “no”, then it asks “Is it smaller than the correct number?” If the answer is “no” then it knows the correct answer is in the lower half of the original set. It then takes the upper half, cuts it in half and repeats the process until it gets the correct answer.

a) Let’s look at the question from above: Chose an integer from 1 to 20. Let’s chose 8.

Cut the set into two smaller sets: 1-10 and 11-20.

1 2 3 4 5 6 7 **8**  9 10 11 12 13 14 15 16 17 18 19 20

Is 10 correct? \_\_\_\_\_\_\_ Is 10 too small? \_\_\_\_\_\_\_\_

b) Since 10 is not too small, the correct answer is between 1 and 10. I cut ten into two groups:

1 2 3 4 5 6 7 **8** 9 10

Is 5 correct? \_\_\_\_\_\_\_\_ Is 5 too small? \_\_\_\_\_\_\_ Now what do I know?

c) What should I do next? Draw the sets below. What number do you think I will try first?

d) How many divisions will it take to find the correct answer? \_\_\_\_\_\_\_\_\_\_\_\_

e) How many divisions would it take if the correct answer was 14? \_\_\_\_\_\_\_\_\_\_\_\_

f) Why do you think this method is used for computing searches?

3) **Hash Search:** Hashing here means breaking a very large set into smaller pieces. It is a combination of the above two search algorithms. What you would do is first assume a “reasonable” search field. In the case of fruit, it would be reasonable to think 100 grams would never be more than 500 calories.

Then you decide how you want to “hash” it. Let’s say into hundreds. You would get 5 sets

1-100 101-200 201-300 301-400 401-500

You would then choose either a linear or a binary search of the five sets. When you found the set that contains the correct answer, you would then do either a linear or binary search of that set. In our case let’s say it is from 101 to 200. You could even do another “hash” on that set and break it into 10’s before you start checking for individual numbers.

a) Try finding 176 by binary and hashing by 10. Which do you think is best? Why?

Binary Hash by 10

101-150 151-200 100 110 120 130 140 150 160 170 180 190

**My Guess is Better Than Yours**

Answer Key

The work of finding the exact numeric answer to a question is often left to a computer. We all know a computer can’t “think”, so how does it find the answer? Below are two methods programmers often use.

1) **Linear Search:** Linear in this sense means “in a line”. Since numbers are generally sorted by size (*in counting it is smallest to largest*), we can have the computer start with the smallest and just check each number in the line until it finds the correct answer. When a number is reached the computer asks, “Is this number the correct answer?” If it gets “yes” then it is done. If it gets the answer “No”, then it moves to the next larger number and asks the question again.

a) If the number chosen is 8, how many questions would the computer have to ask to find the correct answer of 8? \_\_ 8\_\_ What if the number chosen was 14? \_\_­14\_\_\_\_

b) When would this seem like an efficient way to search? When does it seem less efficient?

*If there weren’t a lot of numbers to guess If there were 10,000 numbers even a fast computer would be wasting time.*

c) Why do you think this method is used for computers?

*It seems very easy to program, and a computer can’t think. If the number of choices was small enough it could be done very quickly.*

2) **Binary Search:** Binary here means “cut into two”. The first thing the computer does is cut the largest number in half. This means it now has two smaller sets of numbers. Then asks: “Is this number the correct answer?” If the answer is “no”, then it asks “Is it smaller than the correct number?” If the answer is “no” then it knows the correct answer is in the lower half of the original set. It then takes the upper half, cuts it in half and repeats the process until it gets the correct answer.

a) Let’s look at the question from above: Chose an integer from 1 to 20. Let’s chose 8.

Cut the set into two smaller sets: 1-10 and 11-20.

1 2 3 4 5 6 7 **8**  9 10 11 12 13 14 15 16 17 18 19 20

Is 10 correct? \_\_\_No\_\_\_\_ Is 10 too small? \_\_\_No\_\_\_\_\_

b) Since 10 is not too small, the correct answer is between 1 and 10. I cut ten into two groups:

1 2 3 4 5 6 7 **8** 9 10

Is 5 correct? \_\_\_\_No\_\_\_\_ Is 5 too small? \_\_\_Yes\_\_\_ Now what do I know?

*Use the upper half.*

c) What should I do next? Draw the sets below. What number do you think I will try first?

*Split 6-10 into two sets (actually 3 sets)*

6 7 **8** 9 10

*Try the middle number first.*

*8 is the correct answer. DONE!*

d) How many divisions will it take to find the correct answer?

*Only 3 guesses!*

e) How many divisions would it take if the correct answer was 14?

*See a) above. Then it would need to be 11 -20 split into*

11 12 13 **14** 15 16 17 18 19 20

*Then it would be 11-15 split into three* 11 12 13 14 15.

*Checking 13 would result in having to check 14 and 15.*

*You would find 14 on your next guess.*

*This takes 4 guesses.*

f) Why do you think this method is used for computing searches?

*It takes fewer guesses, especially if there are lots of numbers! Every time the guesses get cut in half.*

3) **Hash Search:** Hashing here means breaking a very large set into smaller pieces. It is a combination of the above two search algorithms. What you would do is first assume a “reasonable” search field. In the case of fruit, it would be reasonable to think 100 grams would never be more than 500 calories.

Then you decide how you want to “hash” it. Let’s say into hundreds. You would get 5 sets

1-100 101-200 201-300 301-400 401-500

You would then choose either a linear or a binary search of the five sets. When you found the set that contains the correct answer, you would then do either a linear or binary search of that set. In our case let’s say it is from 101 to 200. You could even do another “hash” on that set and break it into 10’s before you start checking for individual numbers.

a) Using 101 to 200, try finding 176 by binary and by hashing by 10.

Which do you think is best? Why?

Binary Hash by 10

101-150 151-200 100 110 120 130 140 150 160 170 180 190

Checking gives 151-200 which becomes Linear search gives the 170 Hash with 7 questions,

151-175 176 -200 Binary searches get to the 170 Hash in 2 questions.

Checking gives 176-200 or Now you can use Linear or Binary to search this hash.

176-187 188 189-200 Linear searches would take 7 questions, Binary takes 3.

Checking gives 176-187 or Checking 170-179 or 170-174 175-179

176-181 182-187 gives 175-179

Checking gives 176-181 or Checking 175-179 or 175-176 177 178-179

176-178 179-181 gives 175-176

Checking gives 176-178 or Which gives 176.

176 177 178

Which gives 176

**Just Binary takes 6 guesses Hashing with Linear takes 14 questions**

**Hashing with Binary takes 5**