Computer Science 62
Lab 2

Wednesday, February 17, 2010

The programming exercise for today's laboratory is about the Two Towers problem. To make the problem tractable, we are taking a "bottom up" approach. It is not the way you would normally approach a problem, but it makes sense here because we have already thought about the situation. You will first write an integer subset class and then use it to solve the problem.

IntegerSubsetIterator

For the first step, you are to write a class

```
public class IntegerSubsetIterator
             implements Iterator<ArrayList<Integer>>
```

whose constructor takes a single integer `size` and creates an object that iterates over all the subsets of $\{0, 1, \ldots, \texttt{size-1}\}$. There will be $2^{\texttt{size}}$ subsets.

The iterator will return "subsets" as ArrayLists whose elements are the elements of a subset. For example, running the code

```
IntegerSubsetIterator iter = new IntegerSubsetIterator(3);
while (iter.hasNext())
    System.out.println(iter.next());
```

will produce the output:

```
[]
[0]
[1]
[0, 1]
```

```
[2]
[0, 2]
[1, 2]
[0, 1, 2]
```

Be sure that you understand what the iterator is doing before writing any code. When you are ready, open a new Eclipse project for this lab, and use one of the strategies from the preliminary reading for this lab to implement the class.

The `Iterator` interface requires only three methods, `hasNext`, `next`, and `remove`. The last of these is not used here and can be the "do nothing" method.

Two Towers

(insert Lord of the Rings reference here :)

Now that you have created the subset iterator, you are ready to use it to solve the Two Towers problem. As Bailey describes in his book, one approach is to compute the heights of all the blocks

$$h = \sum_{i=1}^{15} \sqrt{i},$$

and then to look through all subsets of $\{1, 2, \ldots, 15\}$ to find the one whose height comes closest to, without exceeding, $h/2$.

Write a method:

```
public static void twoTowersSolver(ArrayList<Double> blockSizes)
```

that takes as input an ArrayList of block sizes (the easiest way is just to represent a block size by the size of one side). This method should iterate through all of the possible subsets and find the one subset whose sum is closest to $h/2$.

There is one complication: the `IntegerSubsetIterator` class generates subsets of $\{0, 1, \ldots, 14\}$ and you need to correlate these to the actual block sizes. A loop something like:

```
ArrayList<Integer> set = it.next();
```

```
int height = 0;

for( Integer i: set ){
   height += blockSizes.get(i);
}
```

should solve this problem.

Now, try out some different block sizes and print out the difference. For example:

```
public static void main(String args[]){
   ArrayList<Double> blocks = new ArrayList<Double>();

   for( int i = 1; i <= 15; i++ ){
      blocks.add(Math.sqrt(i));
   }

   System.out.println(twoTowersSolver(blocks));
}
```

Have your program print the actual half-height, the height of the best solution, and the blocks in the best solution.

Once you have a working program, experiment by increasing the number of blocks. Try values including 24, 28, and 33. Explain the results and the time it takes to get them.