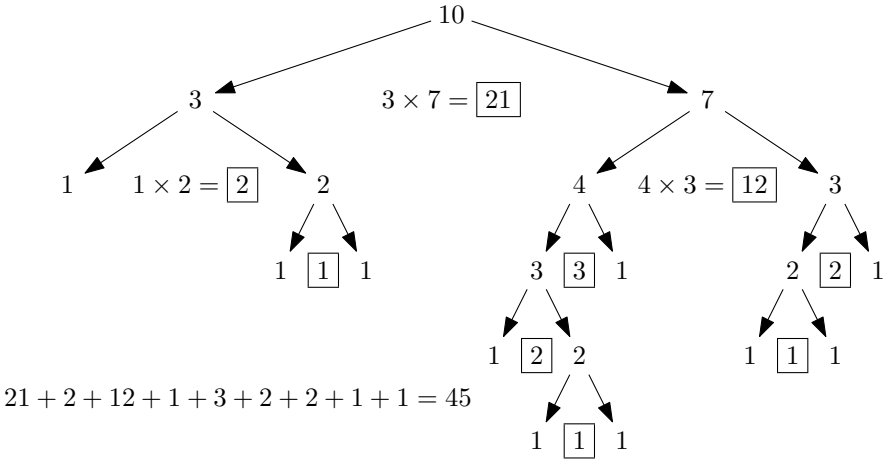
CPSC 320: Tutorial 2

1. Suppose you play the following game. You start with a pile of n stones. You divide the stones into two smaller piles. If the two smaller piles are of size a and b, you write down the product ab and repeat this process on any pile with more than one stone. Eventually, you produce n piles each with one stone. The sum of the numbers you write down is your score. What strategy should you follow to maximize your score? Hint: Try small examples. You should see a pattern. Prove it by induction.

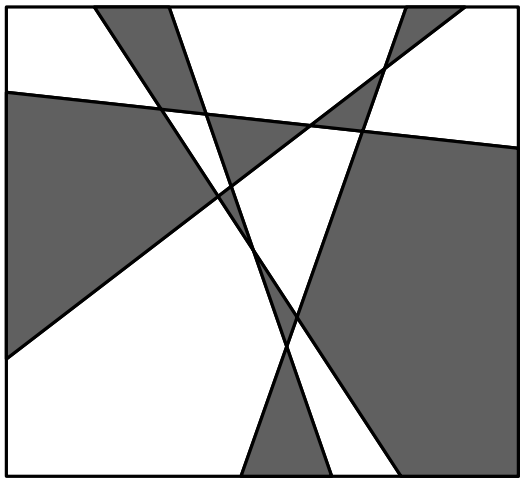
   Here's a big example with 10 stones:



2. Analyze the running time of the following algorithm as a function of n, assuming that each arithmetic operation and comparison takes constant time. Use Θ notation to express your result as simply as possible. Give an informal argument (not necessarily a full proof) to support your answer.

```
Foo(n)
   i = 0
   sum = 0
   while (sum < n)
      i = i + 1
      sum = sum + i
   return i
```

3. Suppose someone draws n (infinitely long) lines on a piece of (infinitely big) paper. The lines divide the paper into regions. Describe an algorithm that colors the regions either black or white so that no two regions with a common boundary are the same color. The following is an example of a good coloring. Hint: Use an input consuming idea.



4. Give a recurrence relation for the running time of the following really bad sorting algorithm.

```
SnailSort(A, p, r)
//
// A is an array, p and r are positions in the array.
//
if A[p] > A[r] then
   exchange A[p] and A[r]
endif

if (p + 1 < r) then
   q := floor ((r - p + 1) / 3)
   SnailSort(A, p + q, r) // sort the last two-thirds
   SnailSort(A, p, r - q) // sort the first two-thirds
   SnailSort(A, p + q, r) // sort the last two-thirds again
endif
```