

Discrete Mathematics, 2016 Fall- Worksheet 11

October 19, 2016

Instructor: Zsolt Pajor-Gyulai, CIMS

In all of the above problems explain your answer in full English sentences.

1. Show algebraically that

$$\frac{(2k+1)(k+1)k}{6} + (k+1)^2 = \frac{[2(k+1)+1][(k+1)+1][k+1]}{6}.$$

2. Let n be a positive integer. Prove the following equations and inequalities by induction.

- (a) $1 + 4 + 7 + \cdots + (3n-2) = \frac{n(3n-1)}{2}.$
- (b) $9 + 9 \times 10 + 9 \times 100 + \cdots + 9 \times 10^{n-1} = 10^n - 1.$
- (c) $1 \cdot 1! + 2 \cdot 2! + \cdots + n \cdot n! = (n+1)! - 1.$
- (d) $2^n \leq 2^{n+1} - 2^{n-1} - 1.$
- (e) $(1 - \frac{1}{2})(1 - \frac{1}{4}) \cdot (1 - \frac{1}{2^n}) \geq \frac{1}{4} + \frac{1}{2^{n+1}}.$

3. Suppose that a grid has $a+1$ vertical lines and $b+1$ horizontal lines. Prove by strong induction that there are $\binom{a+b}{a}$ lattice paths from the lower left to the upper right corner.

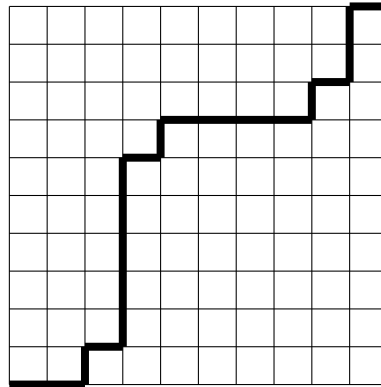


Figure 2: Grid with $a = b = 9$.

4. A flagpole is n feet tall. On this pole, we display flags of the following types: red flags that are 1 foot tall, blue flags that are 2 feet tall, and green flags that are 2 feet tall. The sum of the heights of the flags is exactly n feet. Prove by induction that there are $\frac{2}{3}2^n + \frac{1}{3}(-1)^n$ ways to display the flags.

5. BONUS PROBLEM: Consider the following computer program.

```
function findMax(array, first, last){  
  if (first==last) return array[first];  
  mid = first+ (last-first)/2;  
  a = findMax(array,first,mid);  
  b = findMax(array,mid+1,last);  
  if (a<b) return b;  
  return a;  
}
```

Here `array` is an array of integers. All other variables are integers. We assume that `first` and `last` are between 1 and the number of elements in `array` and that `first` \leq `last`. Note that if `last-first` is odd, then $(\text{last}-\text{first})/2$ is rounded down to the nearest integer.

The purpose of this program is to find the largest value in the array between two indices; that is, it should return the largest value of

`array[first], array[first+1], ..., array[last]`.

Prove that this program fulfills this task.