Homework 12 by Timofei Podlorytov

12.1

the algorithm implementation is in the cpp file provided with.

12.2

a)The solution is in the cpp file.

b)

We have nested for loops which for a triangle with n rows have 1+2+3..+n=(n+1)n/2 iterations. This is equal to $n^2/2+n/2$. For each iteration we have only O(1) complexity since all actions are O(const) meaning we have $O(n^2)$ for the first loop.

Then we search for the maximum in last row taking us n iterations – O(n)
The same number of iterations it takes to track back the path since we have n levels and to print it.

In total we have $O(n^2)+3O(n)=O(n^2)$ -our total time complexity

Now we need to analyze brute force:
Imagine we build a recursion where we return the maximum of left and right path(thus calling recursion). The we call it for the top of the triangle.

For each call we have 2 options meaning for each time we call the function we have 2 recursion calls 1 level – 1 path. For 2 -2 paths and for n levels we would have $2^{n-1}=O(2^n)$ brute force complexity

c)

Greedy:
at each level we choose the larger option thus completing the triangle in linear time. But the fact that the algorithm doesn't work can be proven with counter example:

```
            1
          1   8
        10  1  2
      10 1   1  2
    10  1  1   1  2
```

On first level we would choose 1, then 8, then 2,2,2. Since 1<8, 1<2. BUT. The optimal solution would be to go left and get the sum 32 instead of 15. The algorithm doesn't work.

■