Algorithm Analysis

UNIVERSITY OF CALGARY

CPSC 319 - Data Structures

Benyamin Bashari

Outline

- Analysis Question, not an IQ Test!!
- Length of the longest increasing subarray
- Implementing Matrix Operations

Analysis Question, not an IQ Test!!

- Alice has a mirror, and she is interested in finding out how strong this mirror is.
- Alice lives in a tower with n floors, and we say the strength of a mirror is k if the mirror breaks when Alice throws it from the k-th floor and does not break when Alice throws it from the (k-1)-th floor.
- It is obvious that if the mirror does not break in (k-1)-th floor, it also does not break in floors 1, 2, …, k-2 either.

Analysis Question, not an IQ Test!!

- Assuming Alice has exactly one piece of the mirror, give an algorithm for Alice, in a way that she finds out the strength of the mirror?
  - In worst case, how many floors should Alice visit before she finds out the strength of the mirror?
- What if she has two pieces of the same mirror?
- What if she has many pieces?

Length of the longest increasing subarray

- Basic algorithm:

```
int best = 0;
for(int i = 0 ; i < a.length ; i++) { //start of the subarray
    for(int j = i ; j < a.length ; j++) { //end of the subarray
        boolean sw = true;
        //checking if a[i..j] is increasing
        for(int k = i ; k < j ; k++) {
            if(a[k] > a[k+1]) {
                sw =  false;
                break;
            }
        }
        if(sw)
            best = Math.max(best, j-i+ 1);
    }
}
```

- Analysis? (Just Order)

Length of the longest increasing subarray

- Better approach:
    - Fixing starting point of the subarray (i) and finding the maximum increasing subarray starting from i

```
int best = 1;
for(int i = 0 ; i < a.length ; i++) {//start of the subarray
    //Finding maximum increasing subarray starting from i
    for(int j = i ; j < a.length-1 ; j++) {
        if(a[j] > a[j+1])
            break; //not increasing anymore
        else //a[i..j+1] is increasing
            best = Math.max(best, j+1 - i + 1);
    }
}
```

- Analysis?

Length of the longest increasing subarray

- Best approach:
  - Why i++?!!

```
int best = 1;
for(int i = 0 ; i < a.length ; i++) {//start of the subarray
    //Finding maximum increasing subarray starting from i
    for(int j = i ; j < a.length-1 ; j++) {
        if(a[j] > a[j+1]) {
            i = j; //it will be i++ later
            break; //not increasing anymore
        }
        else //a[i..j+1] is increasing
            best = Math.max(best, j+1 - i + 1);
    }
}
```

- Analysis?

Implementing Matrix Operations

- Arrays and stdin in Java
- Write a class Matrix, with 3 properties
    - n = Number of rows
    - m = Number of cols
    - a = 2D array of elements
- Write 3 operations on them (Consider the conditions)
    - add
    - multiply
    - transpose
- Analyse each of the operations, and test the correctness in TestMatrix.java

Coding Time

- Implement Matrix.java and TestMarix.java