Weekplan: Introduction

Philip Bille

Reading

Introduction to Algorithms, Cormen, Rivest, Leisersons and Stein (CLRS): Chapter 1.

Exercises

1 Find Peak Points Let $A = [2, 1, 3, 7, 3, 11, 1, 5, 7, 10]$ be an array. Solve the following exercises.

1.1 [w] Specify all peak points in A.

1.2 [w] Specify which peak points the two linear time algorithms find.

1.3 Specify the sequence of recursive calls the recursive algorithm produces. First assume the algorithm visits the left half of the array if both directions are valid. Afterwards specify all the possible sequences of recursive calls the algorithm can make when the algorithm can pick any of the two directions when they are both valid.

2 Valleys Give a precise definition of *the valley point problem*.

3 Algorithms and Data Structures

3.1 CLRS [w] 1.1-1.

3.2 CLRS [w] 1.1-2.

3.3 CLRS 1.1-3.

3.4 CLRS 1.1-5.

3.5 CLRS 1.2-1.

3.6 CLRS 1.2-3.

4 Properties of Peak Points Let A be an array of length $n \geq 1$. Solve the following exercises.

4.1 Prove that there always exists at least one peak point in A.

4.2 What is the maximum number of peak points that can be in A?

5 Peak Points Solve the following exercises.

5.1 [†] Implement and test one of the two linear time algorithms for finding peak points.

5.2 [†] Implement the recursive algorithm for finding peak points (be careful not to go out of bounds)

5.3 Describe the worst case inputs for each of the three peak point algorithms.

5.4 [BEng $*$ †] Write pseudo code for an iterative variant of the recursive algorithm for finding peak points. Implement it and test it.

5.5 [BSc] Prove that the recursive algorithm always finds a peak point. *Hint:* Define an appropriate invariant that is valid in each of the recursive calls and use induction.

6 Running Times Solve the following exercises.

6.1 CLRS 1.2-2.

6.2 CLRS 1.1.

7 2D Peak Points Let M be an $n \times n$ matrix (2D-array). An entry $M[i,j]$ is a peak point if it is no smaller than its N,E, S, and W neighbors (i.e. $M[i][j] \geq M[i-1][j]$, $M[i][j] \geq M[i][j-1]$, $M[i][j] \geq M[i+1][j]$ and $M[i][j] \geq M[i][j+1]$). We are interested in efficient algorithms for finding peak points in A. Solve the following exercises.

7.1 Give an algorithm that takes $\Theta(n^2)$ time.

7.2 [$*$] Give an algorithm that takes $\Theta(n \log n)$ time. *Hint:* Start by finding the maximum number in the center column and use this to solve the problem recursively.

7.3 [$**$] Give an algorithm that takes $\Theta(n)$ time. *Hint:* Construct a recursive solution that divides M into 4 quadrants.

M Mandatory Exercise: Fun with Arrays Let A be an array of integers of length n. Look at the following pseudo code and solve the exercises.

```
ARRAYFUN(A, n)
for i = 0 to n-1 do
  for j = 0 to n-1 do
    for k = 0 to n-1 do
      if A[i] + A[j] + A[k] = 0 then
        return true
      end if
    end for
  end for
end for
return false
```

M.1 Explain briefly and concisely what ArrayFun computes.

M.2 Analyze the running time of ArrayFun on an array of length n.

M.3 What happens if "$j = 0$" is changed to "$j = i + 1$" and "$k = 0$" to "$k = j + 1$" in the two inner for loops? Briefly describe what ArrayFun now computes and analyze the running time.