OHJ-2156 Analysis of Algorithms 13.5.2010 Henri Hansen

No calculators or books. Submit your answers on a separate paper.

1. Consider the following algorithm. Assume $A[1...n]$ is an array that contains integers.

```
STUPID(A, n)
  if n = 1 return
  i := n
  k := 1
  while i > 0 do
    A[i] := 0
    i := i - k
    k := k + 1
  endwhile
  STUPID(A, n - 1)
```

   (a) (4 points) Write a recurrence equation for the time consumption of STUPID$(A.n)$, and solve it.

   (b) (2 points) What does the algorithm do to the array A?

   (c) (3 points) Change the algorithm so that it is no longer recursive.

2. Give the Θ, or O and Ω- class of the following functions. Be as accurate as possible.

   (a) (2 points) $\sum_{i=1}^{n} i$

   (b) (2 points) $\log(n!)$

   (c) (2 points) $\sum_{i=1}^{\lfloor \log n \rfloor} i$

   (d) (2 points) $O(n^2) + \Theta(n^3)$

   (e) (2 points) $n^2 \log n + n^2$

3. Solve the following recurrences, assumin $T(n)$ is a constant for $n < 2$. Simply give the answer in Θ.

   (a) (2 points) $T(n) = T(n-1) + 1$

   (b) (2 points) $T(n) = T(n/3) + 3n$

   (c) (2 points) $T(n) = 3T(n/3) + n$

   (d) (2 points) $T(n) = 4T(n/2) + n^2$

   (e) (2 points) $T(n) = T(\log n) + n$

4. There are different coins valued $c_0, \ldots c_n$, and $c_0 = 1p$. It is customary to leave a tip whenever a payment is made. The tip should be the value of the second largest coin used in the payment, when payment is made with as few coins as possible. (If only one type is used, then that is the tip).

   (a) (6 points) Describe the recursive principle and an efficient algorithm that gives you the solution to the minimum number of coins needed to get exact change. Extra points are given, if the solution is such that the tip is minimized.

5. In the following assignment, T_1 is a max-priority queue (max heap implemented in an array) and T_2 is a min-priority queue.

```
DUM(A[1,...,n])
  if n ≤ 2 then return 0
  T1:=∅;T2:=∅
  T1.insert(A[1]);T2.insert(A[2])
  for k:=3 to n do
    if A[k] < A[k−1] do
      T2.insert(A[k])
    endif
    if A[k] > A[k−2] do
      T1.insert(A[k])
    endif
  endfor
  return T1.extractmax()
```

   (a) (4 points) Analyse the time consumption of the algorithm.

   (b) (4 points) Assuming that A is filled with nonnegative integers and that $A[1] = A[2] = 0$, what does the algorithm return? Give an invariant that proves your claim!

6. Think of the following situation: We store a set of numbers dynamically as follows: We have $M[0,\ldots,k-1]$ which is a bit array such that $\sum_{i=0}^{k-1} M[i]2^i$ gives the size of our set. The numbers are stored in some of k arrays, indexed $i = 0,\ldots,k-1$, and A_i has size 2^i. A_i is in use, if $M[i] = 1$, otherwise it is free. Each array in use is always fully used and sorted.

   (a) (4 points) Adding an element to the set happens by finding the first free array A_i, and then moving everything in smaller arrays into it along with the new element. Discuss implementation details and time consumption of the operation.

   (b) (4 points) Searching for an element in the array can be done using a sequence of binary searches. Discuss the time consumption of this operation.

   (c) (4 points) The time consumption analysis of a long sequence of insertions may be overestimated if we simply assume the worst case. Discuss the amortized time consumption of adding n elements to the set.

7. (8 points) (BONUS) Let (V, E) be a directed graph, and S be some set of given vertices. Give an algorithm that checks if there is a path between any two nodes in S. Describe in pseudocode, discuss the data structures used for the graph and set S, and by the algorithm and how they affect the time consumption of the algorithm.

question	1	2	3	4	5	6	7	sum.
max.	9	10	10	6	8	12	8	63
answer								

Note: 55 is the "maximum" of this exam, and it will be graded accordingly. The bonus assignment comes on top of that.