OHJ-2156 Analysis of Algorithms   28.5.2010   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, p, q)
  if p ≥ q return
  i:=p + 1
  while i < q do
     if A[i] < A[p] then
        SWAP(A[i], A[p])
     endif
     if A[i] > A[q] then
        SWAP(A[i], A[q])
     endif
     i:=i + 1
  endwhile
  STUPID(A, p + 1, q − 1)
```

   (a) (2 points) Write a recurrence equation for the time consumption of DUMMY$(A, 1, n)$.

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

   (c) (4 points) Change the algorithm so that it is no longer recursive. Give invariants to both loops in the iterative version.

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

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

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

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

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

   (e) (2 points) $n^2 \log n + n^3$

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

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

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

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

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

   (e) (2 points) $T(n) = T(\sqrt{n}) + \sqrt{n}$

4. Let (V, E) be a directed graph. We wish to represent the graph by either using adjacency matrix or adjacency list. A list requires 32 bits for each vertex and 64 bits for each edge. A matrix, on the other hand, requires only 8 bits for each slot.

(a) (4 points) Assuming you wish to minimize the amount of memory needed for your graph, how does your choice (matrix or list) depend on the number of edges in the graph?

(b) (4 points) We are solving a problem about graph, and the problem can be solved by an algorithm, that looks like the following. (Q is a priority queue.)

```
F(V, E)
   Q:=V
   ...
   while Q ≠ ∅ do
      v:=Q.EXTRACT-MIN()
      for u ∈ V
         if (u, v) ∈ E
            ...
         endif
      endfor
   endwhile
```

Assume that everything that is omitted (marked by $\cdots$) is constant time, and the assignment $Q{:=}V$ is $\Theta(V)$, and Q.EXTRACT-MIN() is $\Theta(\log n)$ when Q contains n elements, and nothing is ever added to Q. Analyse the time consumption for this algorithm for both matrices and lists.

5. In the following assignment, T_1 and T_2 are red-black trees and A is an array. Assume emptying a tree is constant time. $T.(max)()$ returns the largest element in the tree.

```
DUMMY2(A[1, ..., n])
   if n ≤ 1 then return
   T1:=∅
   T2:=∅
   T2.insert(A[1])
   for k:=2 to n do
      if A[k] < A[k − 1] then
         T1.insert(A[k])
         A[k]:=T2.max()
      else
         T2.insert(A[k])
      endif
      k:=k + 1
   endwhile
```

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

(b) (4 points) Argue, using an invariant, that after the program is run, the resulting array A is sorted.

teht.	1	2	3	4	5	yht.
max.	8	10	10	8	8	44
op.						