Exam in course
TDT4120 Algorithms and Data Structures
Tuesday December 9, 2003, 0900–1500

Contact during examination:
Arne Halaas, ph. 41661982; Magnus Lie Hetland, ph. 91851949.
Tools: All calculator types allowed. All printed and handwritten materials allowed.
Please write your answers in the given boxes.
Extra sheets may be submitted, if needed. Write your student number on all sheets.
The exam consists of 8 pages.

Oppgave 1 (5%)

a. Assume that we are comparing implementations of two algorithms, A_1 and A_2, on the same machine. For input size n A_1 uses $9n^2$ steps while A_2 uses $81n \log_2 n$ steps. Each step in A_2 takes twice as much time as each step in A_1. For what values of n will A_1 have a shorter running time than A_2?

Answer (5%):

Oppgave 2 (15%)

Assume that you have three arrays, A, B and C, containing positive real numbers. Each of the arrays has a length n.

a. You want to find a segment $A[i \dots j]$ so that $A[i] \times A[i+1] \times \dots \times A[j]$ becomes as great as possible. How will you do this? Feel free to refer to algorithms in the curriculum. What is the running time?

Answer (5%):

b. You wish to decide whether there exist three numbers a, b and c, such that A contains a, B contains b and C contains c, and such that $a + b + c = x$ for a given x. Describe briefly (either with pseudocode or your own words) an algorithm which solves the problem in $\Theta(n^2 \log n)$, worst-case.

Answer (5%):

c. You wish to solve the same problem as in subtask b, but you can now assume that A, B and C are integer arrays, and that the integers fall in an interval from 1 to M. Describe briefly (either with pseudocode or your own words) an algorithm which solves the problem in $\Theta(n^2)$ time, worst-case. Also give (in brief points) any necessary assumptions about M or the computer/hardware you are using, to make the running time valid.

Answer (5%):

Oppgave 3 (30%)

You have discovered the following pseudocode in an old textbook on algorithms. You are uncertain about which language the book is written in, and you have problems understanding some of the words in the pseudocode:

```
BRILLIG(A[1...N]):
if N = 1:
    return A[1], A[1]
slithy ← ⌊N/2⌋
gyre, gimble ← BRILLIG(A[1...slithy])
wabe, mimsy ← BRILLIG(A[slithy + 1...N])
if gyre < wabe:
    borogroves ← gyre
else
    borogroves ← wabe
if gimble < mimsy:
    mome ← mimsy
else
    mome ← gimble
return borogroves, mome
```

a. What does the algorithm BRILLIG do?

Answer (5%):

b. Assume that $N = 256$. How many comparisons of the type *gyre* $<$ *wabe* and *gimble* $<$ *mimsy* are performed in total? (Your answer should consist of a single number.)

Answer (5%):

c. Write an exact recurrence which expresses the number of comparisons as a function $C(N)$. Assume that $N = 2^M$, for some integer M.

Answer (5%):

d. Solve the recurrence in subtask c. Express the solution exactly, without the use of asymptotic notation.

Answer (5%):

You decide to optimize the algorithm. You change the statement

if $N = 1$:
  return $A[1], A[1]$

to the following:

if $N = 2$:
  if $A[1] < A[2]$:
    return $A[1], A[2]$
  return $A[2], A[1]$

e. Write an exact recurrence expressing the number of comparisons as a function $C(N)$. Assume that $N = 2^M$ for some integer M. The recurrence should also count comparisons of the type $A[1] < A[2]$.

Answer (5%):

f. Solve the recurrence in subtask e. Express the solution exactly, without the use of asymptotic notation.

Answer (5%):

Oppgave 4 (30%)

a. Assume that you have an undirected graph. You know that each node has at most 3 neighbors. Very briefly argue that it is possible to find a two-coloring of the graph such that each node has at most 1 conflict (neighbor with the same color).
Hint: Use the total number of conflicts in your argumentation.

Answer (5%):

Assume that you have given a flow network defined by the following capacity matrix:

$$C = \begin{bmatrix} 0 & 7 & 6 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 0 & 0 & 8 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Here we can see that, for example, the capacity between nodes 4 and 5 is $C[4,5] = 5$. Assume that node 1 is the source and that node 7 is the sink.

b. How many possible cuts are there between source and sink?

Answer (5%):

c. How can one use FORD-FULKERSON to find a minimal cut? Please be brief.

Answer (5%):

d. Find a minimal cut in the flow network. Describe the cut by giving all the nodes that are on the same side as the source.

Answer (5%):

e. A set of paths in a graph $G = (V, E)$ are edge-disjoint if no edge in E occurs in more than one of the paths in the set. Give an algorithm that determines the (maximum) number of edge-disjoint paths between two given vertices s and t of an undirected graph.

Answer (5%):

f. You want to increase the maximum flow of a flow network as much as possible, but you are only allowed to increase the capacity of one edge. How do you find such an edge? (Use pseudo-code or your own words. You may assume the existence of algorithms to compute max flow and min cut.) What is the running time of your algorithm (worst-case, in Θ notation)? Is it always possible to find such an edge? (Justify your answer.)

Answer (5%):

Oppgave 5 (5%)

a. The following problem was given in the exam last year:

> You are given a set S consisting of N real numbers, a real number T and an integer $K \leq N$. Is there a subset Q of S with K elements, where the sum of the elements in Q is at most T?

There exists an algorithm that can solve the problem in $\Theta(N)$ time. Is it reasonable to believe that we can find an equally efficient solution to the following problem? Justify your answer.

> You are given a set S consisting of N real numbers, a real number T and an integer $K \leq N$. Is there a subset Q of S with at most K elements, where the sum of the elements in Q is equal to T?

Answer (5%):

Oppgave 6 (15%)

```
SUM(N)
    top ← 1; S[top] ← N; S[0] ← 2; stacksum ← N
    WRITE('N = ')
    while top > 0
        for i in 1 ... top − 1
            WRITE(S[i], ' + ')
        WRITELINE(S[top])
        while S[top] = 1
            top ← top − 1
            stacksum ← stacksum − 1
        if top > 0:
            S[top] ← S[top] − 1
            stacksum ← stacksum − 1
            while stacksum < N
                top ← top + 1
                if N − stacksum ≤ S[top − 1]
                    S[top] ← N − stacksum
                    stacksum ← N
                else
                    S[top] ← S[top − 1]
                    stacksum ← stacksum + S[top]
            WRITE('  = ')
```

a. What is the output of the function SUM if $N = 6$? Assume that S is an array with as much space as required. Assume that the function WRITELINE outputs its arguments (without spaces between them) and starts a new line, while WRITE outputs its argument (without spaces between them) without starting a new line.

Answer (15%):