UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

COMPUTER SCIENCE THREE

ALGORITHMS AND DATA STRUCTURES

Tuesday 3 September 2002

09:30 to 11:00

Examiners:

Prof. K. Turner (External)

Prof. C. Stirling (Chairman)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

1. (a) The runtime $T(n)$ of many divide-and-conquer algorithms may be described by a recurrence relation of the form

$$\begin{aligned} T(n) &= a\,T(n/b) + \Theta(n^\alpha); \\ T(1) &= \Theta(1), \end{aligned}$$

for constants $a \geq 1$, $b > 1$, and $\alpha \geq 0$, where n is assumed to be a power of b. State a Θ-expression for the function $T(n)$, distinguishing three cases according to the values of α and $\beta = \log_b a$. *[5 marks]*

(b) Suppose x and y are n-bit numbers, where n is a power of 2. Write

$$\begin{aligned} x &= 2^{n/2}x_1 + x_0; \\ y &= 2^{n/2}y_1 + y_0, \end{aligned}$$

where x_0, x_1, y_0 and y_1 are $n/2$-bit numbers. Then

$$xy = 2^n x_1 y_1 + 2^{n/2}(x_0 y_1 + x_1 y_0) + x_0 y_0. \tag{1}$$

Describe a divide-and-conquer algorithm for multiplying n-bit numbers, based on equation (1), in which the "divide" step leads to four recursive subproblems. Write down and justify the recurrence relation describing the runtime of your algorithm. What is the asymptotic solution to this recurrence, according to part (a)? *[8 marks]*

(c) At first sight, evaluating the right-hand side of equation (1) requires four multiplications of $n/2$-bit numbers (discounting multiplications by powers of two, which are just shifts). Demonstrate that, in fact, just three multiplications of numbers of roughly $n/2$ bits suffice. Describe an improved divide-and-conquer algorithm based on this observation, and write down a recurrence relation describing its runtime. Again, what is the asymptotic solution to this recurrence, according to part (a)? *[7 marks]*

[**Hint:** Start with the multiplication $(x_0 + x_1)(y_0 + y_1)$.]

(d) Now consider a proposed a divide-and-conquer algorithm for multiplying n-bit numbers in which x is divided into three $n/3$-bit numbers

$$x = 2^{2n/3}x_2 + 2^{n/3}x_1 + x_0,$$

and y similarly. Suppose the divide step leads to m multiplications of numbers of roughly $n/3$ bits. How small must m be in order to beat the asymptotic runtime achieved in part (c)? Justify your answer. *[5 marks]*

[**Note:** The following inequalities hold: $\log_3 5 < \log_2 3 < \log_3 6$.]

2. (a) Suppose we are given a collection of n items, which for convenience are identified with the numbers $1, 2, \ldots, n$. For each item i a *weight* $w_i \in \mathbb{N}$ is specified. There is also a target weight $t \in \mathbb{N}$. Assume that t is less than, say, a million.

   Design a dynamic programming style algorithm for determining whether there is some subset of the items whose total weight is exactly t. Formally, you are required to determine if there is a vector $x = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ such that $\sum_{i=1}^{n} x_i w_i = t$.

   Your answer should address the following points:

   - What are the subproblems? How are they indexed? *[8 marks]*
   - What is the order in which the subproblems are solved? *[4 marks]*
   - What is the rule for computing a solution to a subproblem in terms of solutions to earlier subproblems? *[6 marks]*

   (b) Suppose,in addition, that each item i has a specified *value* $v_i \in \mathbb{R}^+$. Explain how to modify your solution to part (a) so that it determines the maximum value subset of items whose total weight is equal to the target weight (assuming such a subset exists). Formally, you are required to maximise $\sum_{i=1}^{n} x_i v_i$ over vectors $x = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$, subject to the constraint $\sum_{i=1}^{n} x_i w_i = t$. *[7 marks]*

3. For this question let $G = (V, E)$ be a flow network with capacity function c, source vertex s, and sink vertex t.

   (a) A flow in G is a real valued function $f : V \times V \rightarrow R$ which satisfies the three properties known as *capacity constraint*, *skew symmetry*, and *flow conservation*. Define each of these properties. *[3 marks]*

   (b) Explain what is meant by a cut (S, T) of a flow network. *[2 marks]*

   (c) Recall that for subsets X, Y of V we defined the implicit summation notation by $f(X, Y) = \sum_{x \in X} \sum_{y \in Y} f(x, y)$, and similarly for the capacity $c(X, Y) = \sum_{x \in X} \sum_{y \in Y} c(x, y)$. Show that for any flow f, and any cut (S, T) in G, $|f| \leq c(S, T)$.

   You may use without proof the fact that for any flow f and any cut (S, T), $|f| = f(S, T)$. *[5 marks]*

   (d) Let f be a flow in G, and let G_f be the residual flow graph of f in G. Recall that the capacity in G_f is $c_f(u, v) = c(u, v) - f(u, v)$.

   Show that if G_f contains no augmenting path then there is a cut (S, T) such that $|f| = c(S, T)$. *[7 marks]*

   (e) Suppose $G = (V, E)$ is a directed graph with a source vertex s and sink vertex t. We are interested in finding the maximum k for which there are k vertex-disjoint (and hence also edge-disjoint) paths from s to t. Present an efficient algorithm that solves this problem, and explain why it is correct.

   [**Hint:** A naive application of flow to the graph will fail since it does not necessarily produce vertex-disjoint paths, but something slightly more complex does work.] *[8 marks]*