CS 170, Spring 1994
Final Examination
Professor Manuel Blum

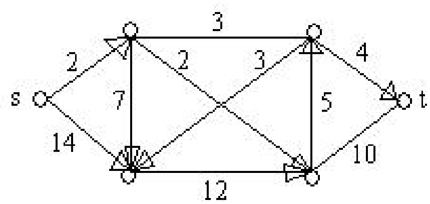
This is a CLOSED BOOK exam.
Calculators ARE permitted.
Do at least 4 of the following 5 problems.
If you do all 5, your grade will be the sum of your best 4 grades.
Try to do all 5 problems.
PUT ALL YOUR ANSWERS IN YOUR BLUE BOOK.

Problem #1a (5 pts)

Is $n^{\log_2 n} = 2^{\log_2 2n}$? If not, is it < or >?

Problem #1b (5 pts)

(i) Find a MAX FLOW in this network:



(ii) Find a min cut in the above network.

Problem #1c (5 pts)

You are given a fair coin. How would you use it to simulate a toss of a (6-sided) die?

Problem #1d (5 pts)

Give an algorithm to multiply 2 complex numbers a+ib and c+id using just 3 real multiplications.

INPUT: 4 real numbers a,b and c,d (denoting a+ib and c+id)
OUTPUT: ac-bd, ad+bc (denoting (ac-bd) + i(ad+bc))

Problem #2a (10 pts)

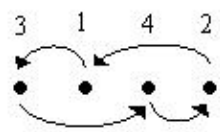
Give an efficient algorithm to determine whether 2 given points $P_1 = (x_1,y_1)$ and $P_2 = (x_2,y_2)$ lie on the same side of a given line, y = ax+b. Here a,b are rational numbers.

Problem #2b (10 pts)

Give an algorithm to find the minumum of n integers $[a_1 .. a_n]$ in O(1) steps on a CRCW parallel computer.

Problem #3 (20 pts)

How many exchanges &lti,j&gt are <u>necessary</u> and <u>sufficient</u> to sort n keys $[a_1 .. a_n]$? The operation &lti,j&gt exchanges a_i with a_j.
<u>HINT:</u> Draw a digraph to represent the desired outcome.
<u>EXAMPLE:</u> [3,1,4,2]



3 exchanges are sufficient.

Problem #4

<u>Polynomial Zero-Finding</u> (PZF) is defined as follows:
<u>INSTANCE</u>: A multi-variable polynomial P(x,y,z,...) with integer coefficients (Example: $3xy^2 - 5x^2z + 7$)
<u>QUESTION</u>: Does the given polynomial have a real root?
i.e. Does P(x,y,z,...) = 0 for (some) any real numbers x,y,z,...? (In above example, answer is <u>YES</u>: x = -7/3; y = 1; z = 0)
The purpose of this problem is to show that SAT(**proportional symbol**)PZF (whence PZF is NP-hard).

Problem #4a (1 pt)

A Karp reduction for SAT(**proportional symbol**)PZF requires a function
f: INSTANCE of __________ --> INSTANCES of __________. (Fill in the blanks.)

Problem #4b (1 pt)

What 3 properties must any such f have?

Problem #4c (8 pts)

The following function (described here by example) <u>almost</u> but doesn't quite work:

f: (x + **y(complex conjugate notation)**) (z + y) (**z(complex conjugate notation)**) --> x(1-y) + zy + (1-z)
Which of the 3 properties <u>does</u> it have, and which <u>not</u>? Give solid (i.e. correct) reasons for your answers.

Problem #4d (10 pts)

Give a function f that works (i.e. has all 3 properties) and prove that it works.

Problem #5 (DYNAMIC PROGRAMMING)

The following problem arises in a video compression scheme:
INPUT: n real numbers $a_1 < ... < a_n$ and a positive integer $k < n$.

OUTPUT: k points (real numbers) $x_1 < ... < x_k$ and a function
f: {1,2, ... ,n} --> {1, ... ,k} that minimizes **summation symbol with terms on top and bottom**($[a_i - x_{f(i)}]^2$).

Problem #5a (4 points)

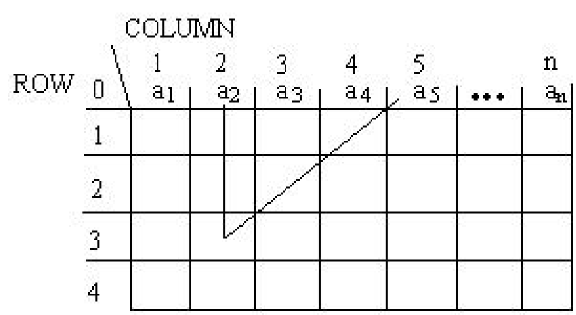
Solve the above problem for k=1. CHECK: If input = [2,4,6,10] and k=1,
then optimal choice of x_1 is 5.5 and **summation symbol**$[a_i - x_1]^2$ = _____ .

Problem #5b (4 points)

Give an efficient algorithm to solve the above problem for k = 2.
CHECK: If input = [2,4,6,10] and k=2, then x_1=4, x_2=10, and **summation symbol**$[a_i - x_{f(i)}]^2$ = ____.

Problem #5c (4 points)

Suppose you are given a table T_{k-1} in which every cell (r,c)
(row = r, column = c) contains the optimal value
min{**summation symbol with terms on top and bottom**$[a_i - x_{f(i)}]^2$} for input $[a_c .. a_{r+c}]$ using k-1 points $x_1,...,x_{k-1}$.



How would you use T_{k-1} to fill T_k ?

Do this for the case [2,4,6,10] by filling in the empty cells in the following tables:

k=1

2	4	6	10
2			
	18.66		

k=2

2	4	6	10
0			
	2		

k=3

2	4	6	10
0			
	0		

Problem #5d (4 points)

Give an algorithm to fill a sequence of n-1 tables, for k=1, 2, ..., n-1.
Your algorithm should show how to use the tables for 1, ..., k-1 to fill the table for k.
(The difference between parts c and d is that c just requires you to fill the above tables, while d requires you to write out the algorithm.

The entry in cell (r,c) of table k should contain

minimum {**summation symbol with terms on top and bottom**$[a_i - x_{f(i)}]^2$ } where the min is over all sets of k points: $x_1, ..., x_k$
& functions f: {1, ..., n} --> {1, ..., k}

Problem #5e (4 points)

How many "steps" does your algorithm take?
