

Exam 1

October 4, 2001

Please write all solutions clearly and legibly in the space provided. A list of available hints is on the chalkboard. If you are given a hint with cost of 3 points on a 10 point problem then the maximum score possible is 7 points.

- (10 pts) Here we look at a nearly completed hand-simulation for the problem of matching m skis with n skiers (for $n \leq m$). Let $h_1 \leq h_2 \leq \dots \leq h_n$ denote the heights of the skiers and $s_1 \leq s_2 \leq \dots \leq s_m$ denote the height of the skis.

Let $c[i, j]$ be the optimal cost for matching skiers $h_1 \dots h_i$ with skis $s_1 \dots s_j$. As proven in the HW 2 solutions,

$$c[i, j] = \begin{cases} \text{output from greedy alg} & \text{if } i = j \\ \min(c[i-1, j-1] + |h_i - s_j|, c[i, j-1]) & \text{if } i < j \end{cases}$$

A secondary array s will hold the choices made. In particular, $s[i, j] = 1$ if the optimal solution for matching $h_1 \dots h_i$ with $s_1 \dots s_j$ matches h_i and s_j . Otherwise $s[i, j] = 0$. Below is the simulation for the input:

skiers $h_1 \dots h_6$: 35 36 45 46 60 65
 skis $s_1 \dots s_{12}$: 30 33 34 40 41 42 43 48 54 59 62 69

		cost of opt sol for $h_1 \dots h_i$ and $s_1 \dots s_j$												1 if h_i and s_j matched 0 otherwise in opt sol for $h_1 \dots h_i$ and $s_1 \dots s_j$													
		j												j													
c		1	2	3	4	5	6	7	8	9	10	11	12	s		1	2	3	4	5	6	7	8	9	10	11	12
		-----												-----													
i	1	5	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	
	2		8	4	4	4	4	4	4	4	4	4	4	2		1	1	0	0	0	0	0	0	0	0	0	
	3			19	9	8	7	6	6	6	6	6	6	3			1	1	1	1	1	0	0	0	0	0	
	4				25	14	12	10	8	8	8	8	8	4				1	1	1	1	1	0	0	0	0	
	5					44	32	29	22	14	9			5					1	1	1	1	1	1			
	6						67	54	46	33				6						1	1	1	1				

Complete the missing entries above (5 in each table). Then use them to output which ski is used by each skier in the optimal pairing represented in s . It is recommended that you briefly show your work so partial credit can be given when appropriate.

2. (35 pts) In each of the following two problems a greedy algorithm is suggested. You are to

- Prove whether or not the given greedy algorithm computes an optimal solution.
- If it does *not*, then give the most efficient algorithm you can to optimally solve the problem. Be sure to prove that your algorithm outputs an optimal solution and analyze its time complexity.

(a) You are given a set $S = \{s_1, \dots, s_n\}$ of n songs where song s_i is ℓ_i minutes long. You want to pick a set $S' \subseteq S$ of songs to place on a single CD which holds 120 minutes such that $\sum_{s_i \in S'} \ell_i$ is maximized under the constraint that $\sum_{s_i \in S'} \ell_i \leq 120$.

Here is a proposed greedy algorithm. Sort the songs so that $\ell_1 \geq \ell_2 \geq \dots \geq \ell_n$. Consider the songs in this order placing each one on the CD if it fits and otherwise not including it. To be sure this is clear, here's pseudo-code.

```
 $S' = \emptyset$   
space = 120  
for  $i = 1$  to  $n$   
    if  $\ell_i \leq$  space  
         $S = S \cup \{s_i\}$   
        space = space -  $\ell_i$   
return  $S'$ 
```

More space for 2a (if needed)

- (b) You are given n unit-duration jobs where job i has an integer deadline time $d_i \geq 0$ and real-valued penalty $p_i \geq 0$. The n jobs can be scheduled at times $0, 1, \dots, n - 1$ with each job scheduled exactly once and only one job can run at a time. If job i is completed by time d_i then there is no cost for it, but if job i completes after time d_i then a penalty of p_i is incurred. Your goal is to find a schedule which minimizes the total penalty.

Here is a proposed greedy algorithm. Sort the jobs so that $p_1 \geq p_2 \geq \dots \geq p_n$. Let the n possible time slots be initially empty where slot i is the unit-length slot that finishes at time i . We consider the jobs in the order $1, 2, \dots, n$. When considering job j , if any of time slots $1, \dots, d_j$ are available then job j is scheduled in the latest such slot. Otherwise, schedule job j in the latest available time slot (from those after d_j).

More space 2b (if needed)

3. (15 pts) You are given as input character strings $U = u_1u_2 \cdots u_n$ and $V = v_1v_2 \cdots v_m$. You can modify U by applying any of the following operations to it: delete a character, insert a character, change a character. Give the most efficient algorithm you can to determine the fewest operations needed to transform U into V . Be sure to prove your algorithm computes the correct answer and analyze its time complexity.

If you need more space go to the back of the exam.

THE CS 441T EXAM ENDS HERE. (If a CS 441T student does the next problem we will note it as extra credit, but no points will be added to your total. Hence CS 441T students should do all they can on problems 1-3 before even considering problem 4.)

THIS PROBLEM IS REQUIRED ONLY FOR CS 539 STUDENTS.

4. (20 pts) Consider the alphabet $\Sigma = \{a, b\}$ where we define “multiplication” by $aa = b$, $ab = b$, $ba = a$ and $bb = b$. Notice that it is not commutative or associative. For example $((ab)a) = a$ whereas $(a(ba)) = b$

Describe the most efficient algorithm you can that takes as inputs a string $x = x_1x_2 \cdots x_n$ of characters of Σ and decides whether it is possible or not to parenthesize x in a way that the value of the resulting expression is a . Be sure to prove that your algorithm outputs the correct answer and analyze its time complexity.

Additional Work for Problem _____

Problem	Points Possible	Points Received
1	10	
2a	35	
2b		
3	15	
4*	20	
total	60 + 20	