

Exam 1

October 12, 2004

- Write your name at the top of this page!
- Please write all solutions clearly and legibly in the space provided. A list of available hints is below. If you are given a hint with cost of 5 points on a 20 point problem then the maximum score possible is 15 points.
- Pace yourself appropriately. I am figuring that you will use about 15 minutes for the first question which leaves about 35 minutes for each of the other two questions.
- Part 3d is only required of CSE 541T students. For 441T students, any points obtained on 3d will be recorded but not added to your exam score. (They'll be treated as extra credit points are for homework.) Thus do not do 3d unless you've done all you can on the rest.

Question 1 Hint (5 pts):

We will fill in the missing value in the s matrix.

Question 2 Hint (10 pts):

We will write the general subproblem form to use for the recursive definition. (Completing part a for you).

Question 3 Hint (10 pts):

We will complete the greedy algorithm. (Completing part a for you).

Problem	Points Possible	Points Received
1	20	
2	40	
3	40 + 5	
total	100 + 5	

1. (20 pts) Here we look at a nearly completed hand-simulation for the problem of constructing an optimal binary search tree where there are n words that are alphabetized where word i is searched for with probability p_i . Recall that for a tree T , $\text{cost}(T) = \sum_{i=1}^n (p_i \cdot d_i(T))$ where $d_i(T)$ is the depth of word i in T where the root is depth 1, the second level is depth 2, and so on.

Let $m[i, j]$ be the expected search cost for the optimal tree for words i to j for $i \leq j$. By definition if $i > j$ then $m[i, j] = 0$ (which makes the recursive definition cleaner). So for example, $m[0, -1] = 0$ since $0 > -1$

Here's the recursive definition for $m[i, j]$:

$$m[i, j] = \begin{cases} 0 & \text{if } i > j \\ p_i & \text{if } i = j \\ \min_{i \leq k \leq j} (m[i, k-1] + m[k+1, j]) + \sum_{\ell=i}^j p_\ell & \text{otherwise} \end{cases}$$

A secondary array, s , will hold the value of k that gave the minimum above. Below all but one entry of each of these arrays is filled in where the row is the value for i and the column is the value for j .

word	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7
p_i	.15	.10	.15	.15	.15	.05	.15	.10
$p_1 + \dots + p_i$.15	.25	.40	.55	.70	.75	.90	1.0

	array m									array s							
	0	1	2	3	4	5	6	7		0	1	2	3	4	5	6	7
0	0.15	0.35	0.7	1.05	1.5	1.65	2.15			0	0	1	2	2	2	2	
1	0	0.1	0.35	0.65	1.05	1.2	1.65	1.95		.	1	2	2	3	3	4	4
2	0	0	0.15	0.45	0.75	0.9	1.35	1.65		.	.	2	2	3	3	4	4
3	0	0	0	0.15	0.45	0.55	0.9	1.2		.	.	.	3	3	4	4	4
4	0	0	0	0	0.15	0.25	0.6	0.8		4	4	4	6
5	0	0	0	0	0	0.05	0.25	0.45		5	6	6
6	0	0	0	0	0	0	0.15	0.35		6	6
7	0	0	0	0	0	0	0	0.1		7

- (a) (10 pts) Compute the values for $m[0, 7]$ and $s[0, 7]$ (the missing entries above). Show enough work that we can see how you got your answer. **If there is a tie then you should break the tie by picking the smallest k among those that give the minimum value.**

- (b) (10 pts) Use the completed s matrix to create the optimal search tree. Remember as a 5 point hint we will fill in the missing entry for you.

2. (40 points) Consider the following problem. You have two trucks each that can hold as much weight as you need. You have a shipment $X = \{x_1, \dots, x_n\}$ of objects where object x_i weighs w_i pounds and further each w_i is an integer between 1 and k . For a subset $X' \subseteq X$ let $W(X') = \sum_{x_i \in X'} w_i$. Your goal is to divide X into two sets A and $\bar{A} = X - A$ so as to minimize $|W(A) - W(\bar{A})|$. That is, you want to partition X into two disjoint sets, one for each truck, so that the weight held by the two trucks is as close to each other as possible.

Use dynamic programming to get an $O(n^2 \cdot k)$ solution to find the minimum difference possible between any partition. (If you can not achieve this time complexity do the best you can). Let me get you started in how to best approach this problem. Think about making use of the following type of yes/no question: Is there a partition of some appropriately chosen items so that the difference in weight is exactly d where you introduce d as a new parameter?

- (a) (10 pts) Write the general subproblem form for your solution. (Remember for a 10 point hint, we will fill in this part.)

- (b) (15pts) Write the recursive definition:

- (c) (3 pts) Where is the the solution to the original problem is located or how it can be computed?

(d) (4 pts) Prove that your recursive definition is correct.

(e) (8 pts) Provide a time complexity analysis for your solution, along with a brief explanation of how you derived it.

3. (40 pts for 441, 45 points for 541) You are given n events e_1, e_2, \dots, e_n where event i requires one unit of time, provides a profit of $p_i \geq 0$ if run, and has a latest legal start time of s_i . (So event e_i can only start at a time t satisfying $0 \leq t \leq s_i$.) Furthermore, each s_i is an integer and you are given that $0 \leq s_1 \leq s_2 \leq \dots \leq s_n$. To further simplify the problem you can assume that all scheduled events run from time $[t, t+1)$ for t some integer. No two events can run simultaneously. The goal is to legally schedule a subset of the events that has maximum profit.

You are to finish designing, prove correct, and describe an efficient implementation for an algorithm based on the following start.

Let $t = s_n$. Let J be the set of events for which t is the last time when they could legally start. Pick a event g in J that has a maximum profit among the events in J . Schedule event g to start at time t .

- (a) (10 pts) Complete the start given above. You can show pseudo-code, or just say, in general (versus just for the first step) what the greedy choice is and then clearly define the subproblem that remains. (Remember for a 10 point hint, we will fill in this part.)

(b) (25 pts) Prove that your resulting greedy algorithm is correct.

- (c) (10 pts) Describe the most efficient way you can to implement your algorithm and analyze the time complexity.

- (d) (5 pts) ONLY DO THIS IF YOU ARE (OR PLAN TO BE) A 541T STUDENT OR YOU ARE A 441T STUDENT WHO HAS DONE ALL YOU CAN ON THE OTHER PROBLEMS. For 441T students these 5 points will not be added to the score but just recorded as extra credit points.

Consider this same problem except events can now run in an interval $[t, t + 1)$ where t is any real. Furthermore, s_1, \dots, s_n can be arbitrary reals (≥ 0). Describe how you would revise your algorithm to handle this and argue why it works.