

Assignment 5 Writeup

1. **Briefly list any parts of your program which are not fully working. Include transcripts or plots showing the successes or failures. Is there anything else that we should know when evaluating your implementation work?**

I am a grad student, and I do not have adds/deletes working for my trie. The substitutions are working perfectly. The plot is attached.

2. **Problem 12-2 from CLRS.**

Insertion:

Each node in the radix tree will have the symbol the node represents, a bit to say whether it is a terminal node, and a pointer to a next node. For each symbol in the key, if the node exists, set the current pointer to that node. Keep doing this until the end of the key is reached, then mark that node as a terminal node.

Search:

Keep a stack and walk the tree in order. When a terminal node is reached, print the stack as a string. Pop off the stack when backtracking. This will output the set S in lexicographical order.

3. **Exercise 15.1-2 from CLRS.**

<i>Pi</i>	10	40	66	80
<i>Di</i>	10	20	22	20

Let P_i equal the profit from cutting at position i , and D_i equal the density of cutting at position i , defined as P_i/i .

Greedy algorithm:

Cut at 3 + Cut at 1 = $66 + 10 = 76$

Optimal algorithm:

Cut at 2 + Cut at 2 = 80

Cut at 4 = 80

Therefore, the greedy algorithm does not solve this problem optimally.

4. **Exercise 15.1-3 from CLRS.**

Because each cost is fixed, we do not have to keep track of a minimum cost array and calculate that along with best. (len-first) is the current position we are cutting at in our loop, and multiplying that by the *fixed* cost c .

Algorithm:

best[0] = 0

for len from 1 to n

 best[len] = $\max_{1 \text{ to } \text{len}} (p_{\text{first}} + \text{best}[\text{len} - \text{first}] - c * (\text{len} - \text{first}))$

return best[len]

5. (Those in CS 858 only) Exercise 15.3–5 from CLRS.

Consider the following profit to cut model:

1	40	66	79
---	----	----	----

Assume we have a limit $l_2 = 1$. Therefore, we can only have one rod of length 2. Also assume that we have a rod of length 6. The optimal way to cut a rod of length 6 is to cut the rod into lengths of 2 three times. However, the limit prevents us from doing so. Therefore, the optimal substructure property does not hold.

6. What suggestions do you have for improving this assignment in the future?

The trie was surprisingly difficult. I would spend longer on the grad student portions in class.