

After Lecture 19 & 20 – Answer any questions on HW5 (due today)

Practice Problems (all taken from previous exams)

1. The step driving the runtime in the solution to the fractional knapsack problem is _____.
 - a) Breaking items into fractions.
 - b) Adding items into the knapsack.
 - c) [Sorting](#).
 - d) Looping through sorted items.
2. How many bits are needed for fixed length encoding if the size of the character set is X ?
 - a) [lg \$X\$](#)
 - b) $X + 1$
 - c) $2X$
 - d) X^2
3. The characters a to h have the set of frequencies based on the first 8 Fibonacci numbers as follows: a : 1, b : 1, c : 2, d : 3, e : 5, f : 8, g : 13, h : 21
A Huffman code is used to represent the characters. What is the sequence of characters corresponding to the following code? 110111100111010
 - a) [fdheg](#)
 - b) [ecgdf](#)
 - c) [dchfg](#)
 - d) [fehgd](#)
4. If a data structure supports an operation FOO such that a sequence of n FOO's takes $O(n \lg n)$ time in the worst case, then the amortized time of a foo operation is (answer A) while the actual time of a single FOO operation could be as low as (answer B) and as high as (answer C).
 - a) $A = n$ $B = \lg n$ $C = n$
 - b) $A = \log n$ $B = 1$ $C = n \log n$
 - c) $A = \log n$ $B = 1$ $C = n^2$
 - d) $A = n$ $B = 1$ $C = n \log n$
5. A pharmacist has W pills and n empty bottles. Let $\{p_1, p_2, p_3, \dots, p_n\}$ denote the number of pills that each bottle can hold. Describe a greedy algorithm, which, given W and $\{p_1, p_2, p_3, \dots, p_n\}$, determine the fewest number of bottles needed to store pills. Prove that your algorithm is correct.
6. How would you modify your pills to bottles algorithm if each bottle also has an associated cost c_i , and you want to minimize the total cost used to store all the pills. Give a recursive formulation of the problem. [We want to find the minimum cost obtainable when storing \$j\$ pills using bottles chosen from the set bottle 1 through bottle \$i\$. This occurs either with or without bottle \$i\$.](#)

7. You are to maintain a collection of items and support the following operations.
- (i) INSERT(item, list): insert item into list (cost = 1)
 - (ii) SUM(list): sum the items in list, and replace the list with a list containing one item that is the sum (cost = length of list)

Use the Accounting Method to show that the amortized cost of an insert operation is $O(1)$ and the amortized cost of a sum operation is $O(1)$. We will maintain the invariant that every item has one credit. INSERT gets 2 credits, which covers one for the actual cost and one to satisfy the invariant. SUM gets one credit, because the actual cost of summing is covered by the credits in the list, but then the result of the sum will need one credit to maintain the invariant. A common error was not putting a credit on the newly created sum.