CECS 528, Learning Outcome Assessment 5 (Version a), March 3rd, Spring 2023, Dr. Ebert

NO NOTES, BOOKS, ELECTRONIC DEVICES, OR INTERPERSONAL COMMUNICATION ALLOWED. Submit solutions to at most 2 LO problems on separate sheets of paper.

Problems

LO1. Which function has a faster big-O growth: $2^{\sqrt{\log n}}$ or $n^{\log^{49} n}$? Defend your answer.

LO2. Recall the use of the disjoint-set data structure for the purpose of improving the running time of the UTS algorithm. For the set of tasks

Task	a	b	c	d	e	f
Deadline	4	4	4	1	4	0
Profit	60	50	40	30	20	10

For each task, show the M-Tree forest after it has been inserted (or at least has attempted to be inserted in case the scheduling array is full). Notice that the earliest deadline is 0, meaning that the earliest slot in the schedule array has index 0. Hint: to receive credit, your solution should show six different snapshots of the M-Tree forest.

LO3. Answer the following with regards to a correctness-proof outline for the Fractional Knapsack algorithm.

(a) Assume $x_1, x_2, \ldots, x_n$ is an ordering of the items in *decreasing* order of profit density (i.e. profit per unit weight). Let f_i denote the fraction of item i that the FK-algorithm adds to the knapsack. Which of the following is a possible sequence for f_1, f_2, f_3, f_4, f_5?

I. 1,1,0.5,0.3,0

II. 0,0,0.4,1,1

III. 1,1,0.3,0,0

IV. 0,0.3,0.5,1,1

(b) Let $f'_1, f'_2, \ldots, f'_n$ be a sequence of fractions that optimizes total profit, and assume that $f_i = f'_i$, for all $i < k$, but $f_k \neq f'_k$. Explain why, in this case, it must be true that $f'_k < f_k$. Hint: in relation to the FKA algorithm, what is the contradiction in case the opposite was true?

(c) For the scenario described in part b, if $f_i = f'_i$, for all $i < k$, but $f_k \neq f'_k$, then what is the relationship between d_k the profit density of x_k, and d_{k+1}, the profit density of x_{k+1}? Explain.

LO4. Given $T(n) = aT(n/3) + n^4$, for some unknown integer $a \geq 1$, what is the least value that a can assume and yet still have $T(n) = \omega(n^4)$. Justify your answer.

LO5. Recall that the `Minimum Positive Subsequence Sum (MPSS)` problem (Exercise 36) admits a divide-and-conquer algorithm that, on input integer array a, requires computing the mpss of any subarray of a that contains both $a[n/2-1]$ and $a[n/2]$ (the end of a_{left} and the beginning of a_{right}. For

$$a = -23, 42, -38, 50, -43, 54, -45, 56, -39, 27$$

(a) Provide the two sorted arrays a and b from which the minimum positive sum $a[i]+b[j]$ represents the desired mpss, for some i in the index range of a and some j within the index range of b.

(b) For the a and b in part a, demonstrate how the minimum positive sum $a[i]+b[j]$ may be computed in $\mathrm{O}(n)$ steps.

Left Sums = -43 7 -31 11 -12

Right Sums = 54 9 65 26 53

Sorted Left Sums = -43 -31 -12 7 11

Sorted Right Sums = 9 26 53 54 65

22 -17 14

10 5 16

-3