CECS 528, Learning Outcome Assessment 6 (Version b), March 10th, 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. Recall the use of the disjoint-set data structure for the purpose of improving the running time of the Unit Task Scheduling UTS algorithm. For the set of tasks

Task	a	b	c	d	e	f
Deadline	3	2	1	0	4	2
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.

LO2. The following pertains to a correctness-proof outline for the Unit Task Scheduling (UTS) algorithm. Let $S = (a_1, t_1), \ldots, (a_m, t_m)$ represent the tasks that were selected by the algorithm for scheduling, where a_i is the task, and t_i is the time that it is scheduled to be completed, $i = 1, \ldots, m$. Moreover, assume that these tasks are ordered in the same order for which they appear in the sorted order. Let S_{opt} be an optimal schedule which also consists of task-schedule-time pairs. Let k be the first integer for which $(a_1, t_1), \ldots, (a_{k-1}, t_{k-1})$ are in S_{opt}, but $(a_k, t_k) \notin S_{\text{opt}}$ because a_k was not scheduled by S_{opt}.

(a) How do we know that S_{opt} must have a task a scheduled at t_k? Hint: what contradiction arises in case time t_k is not being utilized?

(b) What contradiction arises when we assume that a comes before a_k in the UTS ordering? Hint: there are two cases.

LO3. An algorithm has a running time $T(n)$ that satisfies $T(n) = 7T(n/2) + n^2$. Professor Hu is considering an alternative algorithm whose running time $S(n)$ would satisfy $S(n) = aS(n/5) + n^2$ what is the largest value of a that can be used and still have $S(n) = o(T(n))$? Explain and show work.

LO4. 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

$$a = 48, -37, 29, -33, 51, -64, 46, -34, 45, -36$$

(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 $O(n)$ steps.

LO5. Given that $r = ae + bg$, $s = af + bh$, $t = ce + dg$, and $u = cf + dh$ are the four entries of AB, and Strassen's products are obtained from matrices

$$A_1 = a, B_1 = f - h, A_2 = a + b, B_2 = h, A_3 = c + d, B_3 = e, A_4 = d, B_4 = g - e,$$

$$A_5 = a + d, B_5 = e + h, A_6 = b - d, B_6 = g + h, A_7 = a - c, B_7 = e + f,$$

Compute $P_1, \ldots, P_7$ and use them to compute r, s, t, and u.