DUE FRIDAY, APRIL 15, 2022 AT **11:30 pm** VIA GRADESCOPE

C2BASIC(WYZ5RGE)

Collaborators: jhs8cue, spj6s

Sources: Cormen, et al, Introduction to Algorithms. *(add others here)*

PROBLEM 1 *Load Balancing*

You work for a print shop with 4 printers. Each printer i has a queue with n jobs: $j_{i,1}, \ldots, j_{i,n}$. Each job has a number of pages, $p(j_{i,m})$. A printer's workload $W_i = \sum_\ell p(j_{i,\ell})$ is the sum of all pages across jobs for for that printer. Your goal is to *equalize* the workload across all 4 printers so that they all print the same number of total pages. You may only remove jobs from the end of their queues, i.e., job $j_{i,n}$ must be removed before job $j_{i,n-1}$, and you are allowed to remove a different number of jobs from each printer. Give a **greedy algorithm** to determine the maximum equalized workload (possibly 0 pages) across all printers. Be sure to state your greedy choice property.

Solution: Begin by computing each printer's workload. Then equalize their workloads by choosing the printer with the greatest workload and having it print its enqueue job, removing that job from its queue. Repeat this until all four printers have the same workload. The greedy choice property is having the most amount of pages enqueued/having the highest workload. Thus an algorithm that has the "most busy" printer print at each step is a greedy algorithm, as it brings the set of printers closer to being balanced, given the currently known information and available operations.

PROBLEM 2 *Short Answer Questions. (You don't have to explain your answers in your submission, but you should understand the reason behind your answer.)*

A. True or false? Issuing the largest coin first will always solve the *coin change problem* if only two coins are available: the penny and one larger coin. Assume the amount of change is $\geq$ the larger coin.
Solution: True

B. True or false? The *interval scheduling problem* is always guaranteed to have an optimal solution that contains the interval with the earliest finish time.
Solution: True

C. Choose one: In our proof of the correctness of the greedy solution to the *interval scheduling problem*, we exchanged the interval i selected by our greedy choice with another interval that finished earlier/later than interval i. (Your answer is one of "earlier" or "later.")
Solution: Later

D. True or false? A *feasible solution* for the *Huffman encoding problem* is any valid prefix-free code-word table T.
Solution: True

PROBLEM 3 *Optimal Substructure*

Please answer the following questions related to *Optimal Substructure*.

A. What's the difference in how dynamic programming algorithms versus greedy algorithms use *optimal substructure*?

Solution: Greedy algorithms only examine one subproblem and the solutions of said subproblems do not feed into larger problems. The solution to a problem involving dynamic programming requires the solutions to multiple overlapping subproblems through use of memoization.

B. Why did we need to prove the optimal substructure for our greedy Huffman coding algorithm?

Solution: We had to prove the optimal substructure because all greedy algorithms need to have an optimal substructure to work - every greedy choice needs to leave behind a subproblem that is strictly closer to the desired solution.

PROBLEM 4 *Gradescope Submission*

Submit a version of this `.tex` file to Gradescope with your solutions added, along with the compiled PDF. You should only submit your `.pdf` and `.tex` files.