Problem Set 5

mst3k

DATA STRUCTURES AND ALGORITHMS 2 - SPRING 2024

DUE MARCH 13, 2024 AT 11:59P

Collaboration Policy: You are encouraged to collaborate with up to 4 other students, but all work submitted must be your own *independently* written solution. List the computing ids of all of your collaborators in the `collabs` command at the top of the tex file. Do not share written notes, documents (including Google docs, Overleaf docs, discussion notes, PDFs), or code. Do not seek published or online solutions for any assignments. If you use any published or online resources (which may not include solutions) when completing this assignment, be sure to cite them. Do not submit a solution that you are unable to explain orally to a member of the course staff. Any solutions that share similar text/code will be considered in breach of this policy. Please refer to the syllabus for a complete description of the collaboration policy.

Collaborators: list your collaborators

Sources: list your sources

PROBLEM 1 *Goldilocks and the n Bears*

There's been a crime is BookWorld!!

Fictional Detective Thursday Next is investigating the case of the mixed up porridge bowls. Mama and Papa Bear have called her in to help "sort out" the mix-up caused by Goldilocks, who mixed up their n childrens' bowls of porridge (there are n children total and n bowls of porridge total). Each child likes his/her porridge at a specific temperature, and thermometers haven't been invented in BookWorld at the time of this case. Since temperature is subjective (without thermometers), we can't ask the bears to compare themselves, and since porridge can't talk, we can't ask the porridge to compare themselves. Therefore, to match up each bear with it's preferred bowl, Thursday Next must ask the bears to check a specific bowl of porridge. After tasting a bowl of porridge, the bear will say one of "this porridge is too hot", "this porridge is too cold", or "this porridge is just right".

1. Give an algorithm for matching up bears with their preferred bowls of porridge which performs $O(n^2)$ total "tastes". Prove that your algorithm is $O(n^2)$.

   Solution:

2. Give an algorithm which matches bears with their preferred bowls of porridge which performs expected $O(n \log n)$ total "tastes". Intuitively, but precisely, describe how you know the algorithm is $O(n \log n)$.

   Solution:

PROBLEM 2 *Load Balancing*

You work for a print shop with 4 printers. Each printer i has a queue with n jobs: $j_{i,1}, \dots, j_{i,n}$. 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. Note that jobs can only be removed, not added. 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:

PROBLEM 3 *Crossing the Bridge*

There are n people that need to cross a narrow rope bridge as quickly as possible, and each respective person crosses at speeds $s_1, s_2, ..., s_n$ *(note: you can assume these are integers and are sorted in descending order)*. You must also follow these additional constraints:

1. It is nighttime and you only have a single flashlight. One equires the flashlight to cross the bridge.
2. A max of two people can cross the bridge together at one time (and they must have the flashlight).
3. The flashlight must be walked back and forth, it cannot be thrown, mailed, raven'd, etc.
4. A pair walking across together crosses at the speed of the slowest individual. They must stay together!

Describe a greedy algorithm that solves this problem optimally. State the runtime of your algorithm and prove your algorithm always returns the optimal solution. *Note: The obvious greedy algorithm does NOT work here. Be careful! This is more complicated than it appears.*

Solution: