CSE 531: Algorithm Analysis and Design Spring 2025

Homework 2

Instructor: Kelin Luo, Chen Xu **Deadline: Mar/04/2024**

Your Name: _____ Your Student ID: _____

Problems	1	2	3	4	5	Total
Max. Score	8	10	12	10	10	50
Your Score						

Requirement:

- Save and submit your HW 2 submission (Problem 1-4) to "Assignment - HW2" as a single **typed PDF file**. Name your file: Your Last Name Your First Name YourStudents ID Number Assignment Number. Example: **Doe_John_55552222_HW2**
- For HW 2 submission (Problem 5), please submit a single PDF file to "Assignment - HW2Problem5". Example file name: **Doe_John_55552222_HW2Problem5**
- You should view your submission after you upload it to make sure that it is not corrupted or malformed. Submissions that are rotated, upside down, or that do not load will not receive credit. Illegible submissions may also lose credit depending on what can be read. You are responsible for making sure your submission went through successfully.
- To create a figure, you have the option to use software such as "Ipe Drawing Editor" or other similar tools for digital illustrations. Alternatively, you may choose to draw the figure by hand and then insert it into your document.
- Note that your total points for HW2 will not exceed 50. Suppose you receive $P1$, $P2$, $P3$, $P4$, $P5$ points for each of the questions. Your HW2 grade, as counted towards your final grade, will be calculated as $5 * \min\{P1 + P2 + P3 + P4 + P5, 50\}/50$.
- (Only) For HW2 Problem 5, you are allowed (and encouraged) to utilize Generative Artificial Intelligence (GenAI) tools such as ChatGPT, Google Gemini, Meta.AI, Claude.AI, GitHub CoPilot, or others. Proper documentation, citation, and acknowledgment are required when using these tools. Guidelines for citing GenAI in student work at the University at Buffalo can be found through the University Libraries-Citing Generative AI. Please note that generative AI, such as large language models, can produce content that may falsify references, fabricate facts, or inaccurately represent ideas. This content is derived from input provided by human authors and may contain inaccuracies or exhibit biases. Additionally, the legal and copyright status of inputs and outputs from generative AI is currently uncertain. For more information, refer the University Libraries-AI, Authorship and Copyright. You are responsible for the content and accuracy of all work submitted in this class, including any material supported by generative AI.

Problem 1 (8 points). Please use the Algorithm Pseudocode from the Lecture Notes to solve the following two problems.

(a) **(4 points)** Consider an offline caching instance described as follows: we have a cache that can store $k = 4$ pages and the sequence of 16 page requests are $\{1, 5, 2, 6, 4, 1, 6, 5, 3, 2, 1, 4, 7, 5, 2, 6\}$. Please follow the Furthest in Future Algorithm Pseudocode using the Priority Queue in Lecture Notes and present the status of the priority queue after time $t = 10$ and $t = 12$. Refer to the provided example priority queue status after $t = 4$ for guidance.

Pages	**Priority Values**
1	6
5	8
2	10
6	7

Table 1: Priority Queue Status after time 4

(b) **(4 points)** Construct the Huffman code (i.e, please draw the encoding tree and write down the optimum prefix code) for the alphabet {a, b, c, d, e, f, g} with the following frequencies, and provide the weighted length of the code (i.e, please write down the computational formula and calculate the sum over all symbols the frequency of the symbol times its encoding length).

a	b	c	d	e	f	g
1	4	3	3	7	2	5

Problem 2 (10 points). We are given a set of boxes B, where each box i is represented by its dimensions (a_i, b_i), denoting its width and height, respectively. Given are a set of items T, where each item j is represented by its dimensions (a'_j, b'_j), denoting its width and height, respectively. Our objective is to maximize the number of items placed in the boxes. **Note that for any two boxes i and j with $i < j$, we know $a_i \geq a_j$ and $b_i \geq b_j$.** We can put at most 1 item in a box. Item j can be put into box i if one of the following is true:

- $a'_j \leq a_i$ and $b'_j \leq b_i$
- $a'_j \leq b_i$ and $b'_j \leq a_i$

(a) **(2 points)** Given the boxes and items provided below, please provide an optimal packing solution, e.g., put item 1 to box 1, ..., and state how many items can be placed in the boxes.

- 6 boxes of size $\{(6, 5), (5, 5), (4, 4), (4, 3), (2, 3), (2, 1)\}$.
- 6 items of size $\{(1, 5), (5, 3), (2, 3), (1, 3), (4, 4), (4, 6)\}$.

(b) **(8 points)** Please design an efficient greedy algorithm for solving this problem. For this problem, write your algorithm as pseudo code (in fewer than 15 lines), prove the correctness of the algorithm (with safety strategy and self-reduce argument), and describe the runtime in no more than 3 sentences.

Problem 3 (12 points). Consider the problem of scheduling n jobs with different processing times and weights on one machine to minimize the total weighted completion time. That is, we need to try to minimize summation over all jobs $j \in [n]$ the product of the weight and completion time of j.

(a) **(2 points)** For example, we have 5 jobs 1, 2, 3, 4 and 5, and their weights and processing times are as follows:

Jobs	1	2	3	4	5
Processing Times p_j	20	60	15	18	8
Weights w_j	3	6	1	6	2

For example, consider the following schedule:

- We process the jobs in the order 1, 2, 3, 4, 5. The completion times of the 5 jobs are respectively 20, 20+60=80, 80+15=95, 95+18=113 and 113+8= 121. The weighted completion time is

$$20 \times 3 + 80 \times 6 + 95 \times 1 + 113 \times 6 + 121 \times 2 = 60 + 480 + 95 + 678 + 242 = 1555.$$

Your goal is to give the best schedule (or equivalently, the best order) for the 5 jobs. Please provide the **optimal schedule** and compute its weighted completion time.

(b) **(10 points)** State whether the following **two strategies (A) and (B)** in the first step is safe or not. If a strategy is safe, provide a proof; if it is not safe, provide a counterexample to demonstrate why it is not safe.

- Strategy (A): schedule the job with the smallest processing time p_j to be executed first.
- Strategy (B): Schedule the job that has the biggest ratio of weight to length (w_j/p_j) to be executed first.

Problem 4 (10 points). Given are an array A of length n and an integer h. For every integer i in $[h, n]$, let b_i be the h-th largest number in $A[1...i]$. The goal of the problem is to output b_h, b_{h+1}, b_{h+2},..., b_n.

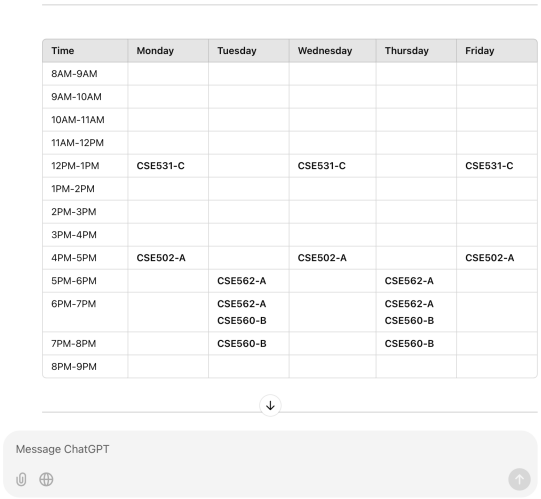
For example, if $h = 4$; $n = 10$ and $A = (50, 80, 10, 30, 90, 20, 100, 40, 65, 70)$. Then $b_4 = 10$, $b_5 = 30$, $b_6 = 30$, $b_7 = 50$, $b_8 = 50$, $b_9 = 65$, $b_{10} = 70$.

Please design an $O(n \log n)$ time algorithm for solving this problem. Please write your algorithm as pseudo code (in fewer than 15 lines), explain its correctness in no more than 3 sentences, and describe the runtime in no more than 3 sentences. (Hint: use the heap data structure.)

Problem 5: (10 points). Consider courses offered in Spring 2025 from the 36 options CSE 5XX, listed as Course-Section: CSE 510-AYY, CSE510-ETH, ... CSE 590-A. Please determine, step by step, the maximum number of courses that can be taken without scheduling conflicts (including time conflicts and course conflicts—for example, selecting CSE 510-AYY means CSE 510-ETH cannot be selected). Provide a visual representation of the optimal course selection. Your response, including written explanations and screenshots, should not exceed 3 A4 pages.

Hints: The courses listed above for Spring 2025 are scheduled either on R (Thursday), TR (Tuesday and Thursday) or MWF (Monday, Wednesday, and Friday). For detailed class time slots (including days: Monday (M), Tuesday (T), Wednesday (W), Thursday (R), Friday (F) and times), please refer to the class schedule on our website. UBCSE-Class Schedule.

Note: You are encouraged to use GenAI tools at any step of solving this problem, such as extracting course schedule information from the course website, outlining algorithm steps to determine the maximum number of courses, creating visualizations. However, be aware that GenAI tools may generate content that includes incorrect references, fabricated facts, or misrepresentations of ideas. You may need to adjust or refine the generated solutions to ensure accuracy. Additionally, make sure to document the entire problem-solving process. See an example reference for illustration[1].



Time	Monday	Tuesday	Wednesday	Thursday	Friday
8AM–9AM					
9AM–10AM					
10AM–11AM					
11AM–12PM					
12PM–1PM	CSE531-C		CSE531-C		CSE531-C
1PM–2PM					
2PM–3PM					
3PM–4PM					
4PM–5PM	CSE502-A		CSE502-A		CSE502-A
5PM–6PM		CSE562-A		CSE562-A	
6PM–7PM		CSE562-A CSE560-B		CSE562-A CSE560-B	
7PM–8PM		CSE560-B		CSE560-B	
8PM–9PM					

Figure 1: This class schedule was created by OpenAI, incorporating course information

(a) **(2 points)** Please fill out the form here(https://forms.office.com/r/t3TjgnnRPB) before completing problem 5(b).

(b) **(5 points)** Please provide a step-by-step approach to determining the maximum number of courses that can be taken without conflicts, along with a visualization of your selected courses.

(c) **(3 points)** Please fill out the form here (https://forms.office.com/r/NDXNkRdhaC) once you have completed 5(b).

[1]OpenAI. Text and Table generated by ChatGPT, Version GPT-4.0. Accessed November 10, 2024. https://chat.openai.com/.