CSCI 332, Fall 2024

Homework 6

Due before class on Tuesday, October 15, 2024–that is, due at 9:30am Mountain Time

---

Submission Requirements

- Type or clearly hand-write your solutions into a PDF format so that they are legible and professional. Submit your PDF on Gradescope.
- Do not submit your first draft. Type or clearly re-write your solutions for your final submission.
- Use Gradescope to assign problems to the correct page(s) in your solution. If you do not do this correctly, we will ask you to resubmit.
- You may work with a group of up to three students and submit ***one single document*** for the group. Just be sure to list all group members at the top of the document. When submitting a group assignment to Gradescope, only one student needs to upload the document; just be sure to select your groupmates when you do so.

Academic Integrity

Remember, you may access ***any*** resource in preparing your solution to the homework. However, you ***must***

- write your solutions in your own words, and
- credit every resource you use (for example: "Bob Smith helped me on this problem. He took this course at UM in Fall 2020"; "I found a solution to a problem similar to this one in the lecture notes for a different course, found at this link: www.profzeno.com/agreatclass/lecture10"; "I asked ChatGPT how to solve part (c)"; "I put my solution for part (c) into ChatGPT to check that it was correct and it caught a missing case.") If you use the provided LaTeX template, you can use the `sources` environment for this. Ask if you need help!

Grading

Remember, submitted homeworks are graded for completeness, not correctness. Correctness is evaluated using homework quizzes.

Each submitted problem will be graded out of six points according to the following rubric:

- Does the solution address the correct problem?
- Does the solution make a reasonable attempt at solving the problem, even if not fully correct?
- Is the presentation neat?
- Is the explanation clear?

- Does the solution list collaborators or sources, or state that the student did not use any collaborators or outside resources?
- Is the solution written in the student's own voice (not copied directly from an outside resource)?

1. Recall the algorithm for the interval scheduling problem. The input to the problem is a set of n jobs with start times $s_1, s_2, \ldots, s_n$ and finishing times $f_1, f_2, \ldots, f_n$.

```
Re-order jobs by finishing time so that f_1 ≤ f_2 ≤ ··· ≤ f_n
S = {}
For j = 1 to n
   If job j is compatible with the jobs in S:
         Add job j to S
   Return S
```

   In this problem, you will analyze the runtime of a naive version of this algorithm, and propose an improvement to get a faster runtime. You should assume that you can sort a list in $\Theta(n\ log n)$ steps in the worst case.

   (a) Suppose that we implement the if statement naively by comparing the new job j to every job in S. For some function $f(n)$ that you choose, give a bound of the form $O(f(n))$ on the running time of this algorithm on an input of size n (i.e., a bound on the number of operations performed by the algorithm).

   (b) For this same function $f(n)$, show that the running time of the algorithm on an input of size n is also $\Omega(f(n))$. (This shows an asymptotically tight bound of $\Theta(f(n))$ on the running time.)

   (c) Propose a different implementation of the if statement (perhaps by keeping track of some additional information) so that the overall algorithm has an asymptotically better running time.

   (d) In the above three steps, we didn't use the words "worst-case", "best-case", or anything else. What sort of running time did we give?

2. Every fall, your family harvests apples at the old family orchard. You have many varied sizes of bins that you harvest the apples into, and a large fleet of identical carts that can all take a total weight of W. Your family has always done it exactly this way:

   - Park a cart at the orchard.
   - As bins of apples are finished, pack them onto the cart in exactly the order they arrive.
   - Once a bin arrives that would put the cart over its maximum weight, send the cart off and start loading the next cart.

   According to this system, we could index the bins with the variable i and write the weight of the i^{th} bin w_i. Because your family members pick apples at different speeds and fill the bins to different levels, the w_i values are varied and unpredictable, and because the apple orchard produces at different levels every year, the total number of bins that will be loaded varies year to year as well.

   Since your family knows that you are taking advanced algorithms, they wonder if you could help them decide whether they might be using too many carts. Maybe they could decrease the number of carts needed by sometimes sending off a cart that was less full, and in this way allow the next few carts to be better packed.

   Prove that, for a given set of apple bins with specified weights, the greedy algorithm currently in use actually minimizes the number of carts that are needed. Your proof should follow the type of analysis we used for the Interval Scheduling Problem: it should establish the optimality of this greedy packing algorithm by identifying a measure under which it "stays ahead" of all other solutions and using a proof by induction to show this.