Questions and Exercises to work out and turn in:

Grading Guidelines:

* A right answer will get full credit when:

1. It is right (worth 25%)
2. It is right **AND** neatly presented making it easy and pleasant to read. (worth an **extra** 15%)
3. There is an **obvious and clear link** between 1) the information provided in the exercise and in class and 2) the final answer. A clear link is built by properly writing, justifying, and documenting an answer (worth an **extra** 60%).
4. Calculation mistakes will be minimally penalized (2 to 5% of full credit) while errors on units will be more heavily penalized.

You are welcome/encouraged to discuss exercises with other students or the instructor. But, ultimately, **personal** writing is expected.

* USE THIS FILE AS THE STARTING DOCUMENT YOU WILL TURN IN. **DO NOT DELETE ANYTHING FROM THIS FILE:** JUST **INSERT** EACH ANSWER **RIGHT AFTER ITS QUESTION/PROMPT**.
* IF USING HAND WRITING (STRONGLY DISCOURAGED), **USE THIS FILE** BY CREATING SUFFICIENT SPACE AND WRITE IN YOUR ANSWERS.
* FAILING TO FOLLOW TURN IN DIRECTIONS /GUIDELINES WILL COST **A 30% PENALTY.**

Objectives of this assignment:

* to use and manipulate the concepts presented in this module
* to propose and write algorithms in pseudocode
* to analyze the time complexity of algorithms
* to analyze the space complexity of algorithms
* to learn autonomously new concepts

What you need to do:

Answer the questions and/or solve the exercises described below.

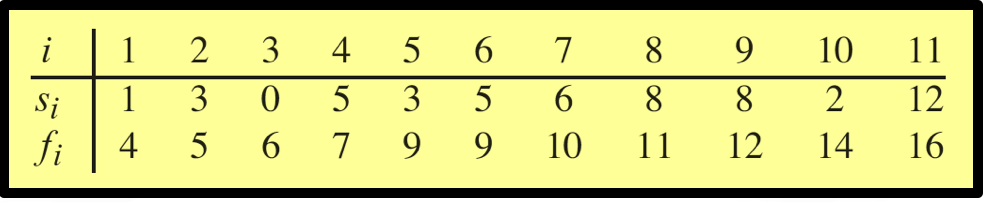
Exercise (100 points)

Sally proposes a different algorithm to the activity-selection problem: instead of always selecting the first activity to finish, she proposes to select the last activity to start that is compatible with all previously selected activities. Let us call this algorithm *Sally's Algorithm*.

1. (15 points) Describe how *Sally's Algorithm* is a greedy algorithm. Just using common sense, explain how this heuristic “makes sense”. (**Hint**: review in the lecture or on the textbook how the algorithm we studied is justified. Inspire and follow the same path to make your case)

Sally’s Algorithm is a greedy algorithm because at each step it makes the locally optimal choice, it chooses the last activity to start that doesn’t conflict with the already selected activities. This is based on the heuristic that by starting as late as possible, it leaves room for as many other activities as possible before it. This “makes sense” as a heuristic because it minimizes the wasted space at the end of the schedule.

1. (30 points) Execute *Sally's Algorithm* on the same problem studied in class with the same starting and finish times:



Explain in **detail** only how you determine the **first two** activities you select. Provide the set of activities *Sally's Algorithm* selects.

Since Sally’s Algorithm proposes to select the last activity to start that is compatible with all previously select activities, we must first start with sorting the start time. Currently, the activities are sorted based on finish time since it was based on selecting the first activity to finish. So now we must adjust the sort of the starting time in increasing order to adjust for Sally’s Algorithm. Therefore, the new sort is a3, a1, a10, a5, a2, a4, a6, a7, a9, a8, a11. We start by first choosing the activity with the latest start time among the compatible activities, which is a11 since it has the start time of 12. Then we need to consider the next activity that finishes before a11 starts. In the new sorted order, the second activity selected is now a8 since it has 8 as the start time and it does not overlap with a11. The full set S of activities Sally’s Algorithm selects is {a11, a8, a4, a2}.

1. (40 points) Execute the iterative algorithm GREEDY-ACTIVITY-SELECTOR(s,f) (See Textbook p.421) on the same problem above. Explain in detail only how you determine the first two activities you select. Provide the set of activities GREEDY-ACTIVITY-SELECTOR(s,f) selects. Compare and discuss the sets of activities found by *Sally's Algorithm* and GREEDY-ACTIVITY-SELECTOR(s,f).

The iterative algorithm GREEDY-ACTIVITY-SELECTOR(s,f) assumes that the input activities are already ordered by increasing finish time. Line 2 initializes A to {a1}, because a1 must be chosen. F[k] is always the maximum finish time of any activity in A. The for loop of Line 4 to 7 finds the compatible activity that finish earliest. So the first compatible activity will be a1 since it already sorted so will have the earliest finish time. Next, we compare s[2] to f[1]. s[2] is 3, which is less than 4 so we go to the next item. s[3], which is 0, is also less. The next is s[4] which is 5. This is greater that the finish time for a1, so a4 will be our next activity. The final set after the algorithm completes is {a1, a4, a8, a11}. The sets of activities found by Sally’s Algorithm and GREEDY- ACTIVITY- SELECTOR(s,f) are very similar. They are both are greedy algorithms since at each step they both makes the locally optimal choice that doesn’t conflict with the already selected activities. They also both yield essentially the same results but in opposite order.

1. (15 points) **Prove** that it yields an optimal solution. Insure to follow the same steps used in the lecture to show that the greedy approach to select the earliest finish time activity that is compatible does deliver an optimal solution. Define well your notations just like the lecture (or textbook).

To prove that it yields an optimal solution, we follow Theorem 16.1 in which we consider any nonempty subproblem Sk, and let am be an activity in Sk with the earliest finish time. Then am is included in some maximum-size subset of mutually compatible activities of Sk. Here, since am has the earliest finish time in Sk, then fm  fj for all activities aj  Sk. Let aj Ak with aj having the earliest finish time in Ak. The activities Ak are disjoint by assumption, it is a maximum-size subset of mutual activities in Sk. aj is the first activity in Ak to finish. fm fj. In addition, | | = | |. Then, is a maximum-size subset of mutually compatible activities of (i.e., includes ). This proves that the algorithm yields an optimal solution.

What you need to turn in:

* Electronic copy of this file (including your answers) (standalone). Submit the file as a Microsoft Word or PDF file.
* Recall that answers must be well written, documented, justified, and presented to get full credit.
* How this assignment will be graded:
* A right answer will get full credit when:
* It is right (worth 25%)
* It is right AND neatly presented making it easy and pleasant to read. (worth 15%)
* There is an obvious and clear link between 1) the information provided in the exercise and in class and 2) the final answer. A clear link is built by properly writing, justifying, and documenting an answer (worth 60%).
* Calculation mistakes will be minimally penalized (2 to 5% of full credit) while errors on units will be more heavily penalized.
* You are welcome/encouraged to discuss exercises with other students or the instructor. But, ultimately, personal writing is expected.

**Appendix**: Grading: What is an OBVIOUS and CLEAR LINK?

Here is an example to explain what an **obvious and clear link** is and how we grade your work.

Consider the following problem:

"(100 points) John travels from Auburn to Atlanta in his car at a speed of 60 mph. Leaving at 8am, at what time will John reach Atlanta".

Here are the answers of three students and their scores:

* **Student 1** answers: "9:48am". Student 1 will get 25 points.
* **Student 2**answers: "John will reach Atlanta at 9:48am". Student 2 will get 25+15 = 40 points
* **Student 3** answers: "The time t to travel a distance d at speed v is equal to d/v = d/60mph. The problem does not provide the distance d from Auburn to Atlanta. Based on GoogleMaps, the distance from Auburn to Atlanta is approximately 108 miles (**document is attached**).



Therefore, the time t = 108 miles/60mph \* 60 minutes/hour= 108 minutes. Since John left at 8am, he will then reach Atlanta at 8am + 108 minutes = 8 am + 60 minutes + 48 minutes = 9:48".

**Student 3** will get 25 + 15 + 60 = 100 points

Do you see the **direct** **link** going from the data provided in the question to the final answer, using general knowledge/formula and documents?.... Can you now solve the following problem and get 100 points?

"(100 points) Alice travels from Auburn to Atlanta in her car at a speed of 60 mph. Leaving at 8am, at what time will Alice reach Atlanta assuming that she had a flat tire that delayed her 30 minutes".