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)

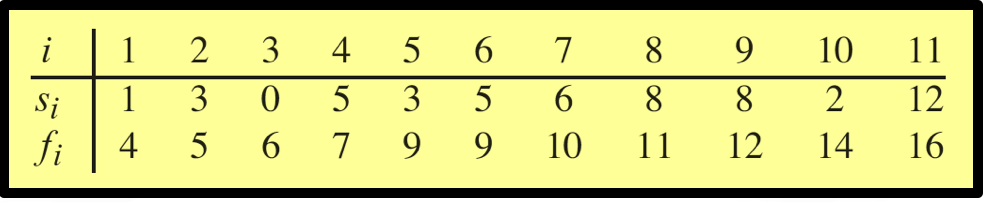
Sally proposes a different algorithm to the activity-selection problem: instead of always selecting the first activity to finish, she proposes to select the last activity to start that is compatible with all previously selected activities. Let us call this algorithm *Sally's Algorithm*.

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)

The idea is we are always going to take advantage of the next best ai given how the loop invariant will work for “Sally’s Algorithm”. Sally proposes that we start from the last possible activity instead of the usually first available one. This is also the idea of going in reverse alphabetical order in a sense. The two other heuristic ideas that make this a valid solution still are the fact by ensuring each selected activity is not overlapping one another. This is further proven by the notion that as long as that the latest finishing time finishes first that this will leave as much space for the remaining “activities” to select as many together as possible. Then you are left with the fact this algorithm will build a solution adding one activity at a time from an empty

set A = {} always selecting first the activity that gives the most benefit in relation to the loop invariant of always selecting the latest finishing time first.

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.

First the algorithm will work to set the set S = {} to an empty set that represents an arbitrary optimal set. The first activity that will be selected is the activity with the latest finish time. Since the array starts by getting sorted in non-decreasing order, the last activity in the array will be used to initialize the first activity to the list with a starting time of 12 and finishing time of 16.

Set after the first activity selection S = {11}

For the second activity selection Sally’s algorithm will again select the latest possible end time without overlapping any of the activity times. With that said the algorithm will skip over activity 10 since the start time of 2 is much after the start time of 11. When we move to the next activity point 9, it has a start time of 8 and a finish time of 12 which equals the start time for activity 11. This means that activity 9 is “safe” to add to the list.

Set after the second activity selection S = {9, 11}

The following selections did not require any explanation:

3rd Selection: S = {8, 9, 11}

4th Selection: S = {7, 8, 9, 11}

5th Selection: S = {4, 7, 8, 9, 11}

Final Selection: S = {1, 4, 7, 8, 9, 11}

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).

In a sense the GREEDY-ACTIVITY-SELECTOR will work in the opposite fashion as “Sally’s Algorithm” did. It will still start with an arbitrary empty set S = {}, but it will initialize the first selection with the first sorted activity in the total set. This means after the first activity selection the set will be:

Set after the first activity selection S = {1}

For the second activity selection, GREEDY-ACTIVITY-SELECTOR will select the next activity that starts after the finish time of the previously selected activity. With that said, the algorithm will check activity 2, which starts at 3 and finishes at 5, but it is not compatible since its start time is before the finish time of activity 1. The algorithm continues to check activity 3, which starts at 0 and finishes at 6, but it is also not compatible. Activity 4, which starts at 5 and finishes at 7, is compatible and will be added to the set.

Set after the second activity selection S = {1, 4}

The following selections did not require any explanation:

3rd Selection: S = {1, 4, 6}

4th Selection: S = {1, 4, 6, 8}

Final Selection: S = {1, 4, 6, 8, 11}

Both algorithms look to maximize the number of non-overlapping activities, but they follow different heuristics. Sally's Algorithm selects the latest starting compatible activity possible, which often leads to selecting more activities that start later. In contrast, GREEDY-ACTIVITY-SELECTOR chooses the earliest finishing compatible activity, which tends to favor activities that start earlier. The difference in the heuristic approaches results in different sets of activities being selected, with Sally's Algorithm finding a slightly larger set of compatible activities in this case.

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 these algorithms yield an optimal solution, let’s first point out the fact that will only produce a version of an optimal solution. It does not prove that it is the most optimal solution. First to match the notation that was covered through the lectures and the book. Let’s let Sk represent the set of activities that will start after activity ak finishes. Since ak has the earliest finishing time in Sk it is chosen as the first activity to be safely added to the set. Now, consider a maximum-size subset Ak of mutually compatible activities in Sk. Then if ever ak = am then the set has been exhausted and we are done. For the second case for when ak is not equal to am then we will be a new set that is the prime subset A’k = Ak – {am} union {ak}. This then gives and proves that A’k is a maximum-size subset of mutually compatible activities in Sk that also includes ak. This proves that the greedy approach yields AN optimal solution. This does not mean though that it is the only optimal solution or even the best optimal solution. But simply proves that it will become one following the structure of the greedy algorithm.

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".