CSC 2420 Fall 2023, Assignment 1
Due date: Wednesday, October 11 at 11 AM,)

It is certainly preferable for you to solve the questions without consulting a published source. However, if you are using a published source then you must specify the source and you should try to improve upon the presentation of the result or at least express the result in your own words.

If you would like to discuss any questions with someone else that is fine BUT at the end of any collaboration you must spend at least one hour playing video games or watching a Blue Jays game or maybe even start reading a good novel before writing anything down.

If you do not know how to answer a question, state "I do not know how to answer this (sub) question" and you will receive 20% (e.g. 2 of 10 points) for doing so. You can receive partial credit for any reasonable attempt to answer a question BUT no credit for arguments that make no sense.

In class or on Piazza, I can clarify any questions you may have about this assignment or any material in the course. Do not spend too much time on any question.

1. This question concerns the makespan problem on $m$ identical machines.

   - [10 points]

     Argue for $m = 2$ (resp. $m = 3$) machines that *any* (not necessarily greedy) deterministic online algorithm would have competitive ratio no better than $\frac{3}{2}$ (resp. $\frac{5}{3}$) so that the natural greedy online algorithm approximation is tight for $m = 2$ and $m = 3$ for any online algorithm. Note that when considering negative results for deterministic algorithms, the adversary can stop the nemesis sequence at any time.

   - [5 points]

     Consider the nemesis sequence (i.e. $p_i = 1$ for $1 \leq i \leq m(m-1)$ and $p_{m(m-1)+1} = m$) that forces the ratio $2 - \frac{1}{m}$ for Graham's greedy algorithm. Consider the same sequence for the makespan problem on $m$ machines but now in the random order model (ROM). Provide a "good" estimate for the expected competitive ratio of Graham's greedy algorithm when executed on this set of inputs? Note: the expectation is wrt to the randomness in choosing the sequence uniformly at random. Your argument can just give good intuition but of course a proof is better.

     **Hint:** The ROM model is equivalent to randomly choosing a time $y_i \in [0, 1]$ for the $i^{th}$ item and then the ordering on the input jobs is defined by the ordering of the $y_i$ variables. What is the expected location of the "big" item (i.e. $p_j = m$) in the random ordering?

   - [5 points]

     Can you modify the above sequence to force a $2 - \epsilon$ lower bound on the expected competitive ratio for Graham's greedy algorithm in the ROM model ? More precisely, show that for all $\epsilon$, there exists a sufficiently large $m$, such that there is an input sequence forcing the competitive ratio to be $\geq 2 - \epsilon$.

2. The following question involves the {0,1} knapsack problem. See slide 8 of the week 2 slides for notation.

   - [10 points]

Show that no deterministic (fixed or adaptive) priority algorithm (without the ability to revoke) can achieve a constant approximation ratio for the general knapsack problem. Assume without loss of generality that the knapsack has size 1.
Hint: Consider 3 types of items, up to at most $n$ items of each type. Type 1 items have value = 1 and size = 1, type 2 items have value = $\frac{1}{\sqrt{n}}$ and size $\frac{1}{n}$, and type 3 have value = $\frac{1}{n}$ and size $\frac{1}{n^2}$. Note (as in deterministic online algorithms) that the adversary can stop the input sequence at any time.

- [5 points]

  Show that there is a deterministic adaptive priority algorithm with revoking that acheives a $\frac{1}{2}$ aproximation ratio.

- [5 points]

  Show that there is a simple $\frac{1}{4}$ approximate (barely) randomized priority algorithm (without the ability to revoke)?.

- [5 points]

  Is there a better than $\frac{1}{4}$ approximation for a randomized priority algorithm (without revoking)?

- Bonus question: What is the best randmized priority algorithm for the knapsack problem? More specifically, what is the best negative rasult?

3. [10 [points]

   Consider the $k$ set packing problem which assumes that all sets $S_i$ have the same size $k$. Show that the natural greedy algorithm (which sorts the sets so that $w_1 \geq w_2 \geq w_n$) provides a $\frac{1}{k}$ approximation. Hint: construct a $k-1$ charging function.