Data Structures and Algorithms
Homework Assignment 5

Given: February 15, 2018 **Due:** February 21, 2018

Note: The homework is due **electronically on Gradescope and Canvas** on Wednesday, February 21 by 11:59 pm EST. For late submissions, please refer to the Late Submission Policy on the course webpage. You may use a maximum of 2 late days on this homework.

A. **Gradescope**: You must select the appropriate pages on Gradescope. Gradescope makes this easy for you: before you submit, it asks you to associate pages with the homework questions. Failing to do so will get you points off, which cannot be argued against after the fact. Gradescope may prompt you with a warning to select your cover page, please ignore this warning.

B. **LaTeX**: You must use the `hw121.cls` Latex template provided on the course website, or a harsh penalty will be incurred. Handwritten solutions or solutions not typeset in Latex will not be accepted.

C. **Solutions**: Please write concise and clear solutions; you will get only a partial credit for correct solutions that are either unnecessarily long or not clear. Please refer to the Written Homework Guidelines for all the requirements.

D. **Algorithms**: Whenever you present an algorithm, your answer must include 3 separate sections:

1. A precise description of your algorithm in English. No pseudocode, no code.
2. Proof of correctness of your algorithm
3. Analysis of the running time complexity of your algorithm

E. **Collaboration**: You are allowed to discuss **ideas** for solving homework problems in groups of up to 3 people but *you must write your solutions independently.* Also, you must write on your homework the names of the people with whom you discussed. For a clarification on the collaboration policy, please see Piazza @547

F. **Outside Resources**: Finally, you are not allowed to use *any* material outside of the class notes and the textbook. Any violation of this policy may seriously affect your grade in the class. If you're unsure if something violates our policy, please ask.

1. [20pts - Heap Insertions] Show that, for any n, there is a sequence of insertions in a max-heap that requires $\Omega(n \lg n)$ time to process.

2. [20pts - Ben's Whale Watching Adventure] Sad he didn't win the DVD during "Palentine's Day", Ben decided to spend the rest of his day researching the best whale watching companies on TripAdvisor. His pals told him they would consider going with him if he presents several highly rated options to them and lets them decide which one to choose. Therefore, Ben decided to create a max-heap W to add all n companies he researched, setting the key as the rating the company had on TripAdvisor.

Ben's pals want to have a good time, so they don't want Ben to send them any option with a rating less than r (which is not necessarily in W). Give an $O(k)$ algorithm to help Ben compile a list of all options rated better than r, where k is the number of such options.

For example, if Ben found the 5 companies $(c_1, c_2, c_3, c_4, c_5)$ with ratings $(3.5, 4.9, 3.6, 2.7, 1.1)$ and Ben's pals only want trips with a rating better than 3.5, then Ben should give them the list $\{c_2, c_3\}$.

3. [30pts - Pierogi Customer Line] After Jess's pierogi restaurant became an instant success, she is overwhelmed by the number of visitors to her restaurant. She decided on a new system of lining up for her customers and wants her line to have the following functionalities:

- JoinLine(c): Direct a new customer c to the end of the line.
- RemoveCustomer(): Remove and return the customer at the end of the line
- ServeCustomer(): Remove and serve the customer at the front of the line, and return them.

Help Jess implement her new lining system with *three* stacks (where she is only allowed to access the component stacks through the functions Push and Pop) and O(1) additionally memory. To minimize customer wait time, the amortized time for each operation has to be O(1). Each customer has to be stored in exactly one of the three stacks. If the line has no customer, you can just return `null` for RemoveCustomer and ServeCustomer.

4. [30pts - Pierogi Supermarket Frenzy] While her customers are busy lining up, Jess makes a frantic visit to "Pierogi Factory," a supermarket that has all the ingredients she could ever need, since her restaurant was running low on supplies. Each of the n ingredients at this supermarket has a distinct, positive, integer price, which you can find in $P[1..n]$.

A. Jess wants to know if she can buy a nonempty subset B of ingredients with prices listed in P with less than $|B|^2$ dollars. She doesn't want more than one of each ingredient. Help her come up with an $O(n \lg n)$ algorithm, to determine if such a subset exists.

B. Jess changed her mind. She now is set on buying m ingredients, where $1 \leq m \leq n$. Jess wants to know if she can buy a subset B', where $|B'| = m$, of ingredients with prices listed in P with less than m^2 dollars. She still doesn't want more than one of each ingredient. Design an $O(n)$ algorithm to determine if this is possible. (Note: This is the same question as above, but here you are being asked to consider subsets of a particular cardinality).

REMINDER: Don't forget to submit to Canvas as well!