**COMP 5711: Advanced Algorithm**
**2018 Fall Semester**
**Midterm Exam Solutions**

Problem 1 (20pts) We can use a linked list as the underlying data structure. The operations are as follows.

- `INSERT`$(S, x)$: Insert $x$ to the linked list. The cost is $O(1)$.
- `DELETE-LARGER-HALF`$(S)$: Find the median of elements in the linked list using the linear-selection algorithm. Scan the linked list and delete the large half one by one according to the median. The cost is $O(|S|)$.

We use the accounting method. Charge each INSERT operation an amortized cost of $O(1)$, which pays for the actual cost of the INSERT operation, and stores $O(1)$ as credits. Half of the elements are deleted in DELETE-LARGE-HALF operation. The larger-half has credits $\Omega(|S|)$ which is enough to pay for the $O(|S|)$ cost of DELETE-LARGE-HALF.

Problem 2 (30pts) The operations are as follows.

- `ENQUEUE`$(Q, x)$: Push $x$ into the $S_1$.
- `DEQUEUE`$(Q)$: If $S_2$ is not empty, pop from $S_2$. Otherwise, pop each element from $S_1$ and push it immediately into $S_2$. Then pop from $S_2$.

Define the potential for a queue as 2 times the number of elements in the $S_1$. So we have $\phi D_0 = 0$ and $\phi D_i \geq 0$. For `ENQUEUE` operation, the potential change is 2 and the actual cost is 1. So the amortized cost of `ENQUEUE` is $O(1)$. For `DEQUEUE`, the potential decreases by $2|S_1|$ and the actual cost is $2|S_1|$. So the amortized cost of `DEQUEUE` is also $O(1)$.

Problem 3 (30pts) We independently and uniformly sample (with repetition) $k = \frac{2}{\delta} \ln m$ students from $S$. And let $C$ be the set of sampled students. We show that, with probability $1 - \frac{1}{m}$, $C \cap G_i \neq \emptyset$ for all $i \in [1, m]$. Let $s_1, \ldots, s_k$ be the students we sampled. We fixed some $i$ in $[1, m]$. Since all the students are sampled independently,

$$Pr[G_i \cap C = \emptyset] = Pr[(s_1 \notin G_i) \wedge \cdots \wedge (s_k \notin G_i)] \leq \prod_{j=1}^{k} Pr[s_j \notin G_i].$$

$G_i$ has at least $\delta n$ members, so for any $j \in [1, k]$,

$$Pr[s_j \notin G_i] \leq \frac{n - \delta n}{n} = 1 - \delta.$$

Recall that $k = \frac{2}{\delta} \ln m$. We have

$$Pr[G_i \cap C = \emptyset] \leq \prod_{j=1}^{k} Pr[s_j \notin G_i] = (1-\delta)^{\frac{2}{\delta} \ln m} \leq \left(\frac{1}{e}\right)^{2 \ln m} = \frac{1}{m^2}.$$

Finally, by union bound, we obtain

$$Pr\left[(C \cap G_1 = \emptyset) \wedge \cdots \wedge (C \cap G_m = \emptyset)\right] \leq \sum_{i=1}^{m} Pr[G_i \cap C = \emptyset] \leq m \cdot \frac{1}{m^2} = \frac{1}{m}.$$

Problem 4 (20pts) We call a pivot "good" if it is between the 25% and 75%-perentile of the current elements, otherwise "bad". The probability of getting "good" pivot is then 1/2. Note that each "good" pivot reduces the number of elments by at least 1/4. So the number of "good" pivot is $O(\log n)$. Let $X_i$ denote the number of pivots between the $i$-th good pivot (not including) and the $(i+1)$-th "good" pivot (including).

$$E[X_i] = \sum_{j=0}^{\infty} j \times Pr(X_i = j) \leq \sum_{j=0}^{\infty} \frac{j}{2^j} = 2.$$

According to the linearity of expectations, the expected number of iterations (pivots) is $E[\sum_{i=1}^{O(\log n)} X_i] = \sum_{i=1}^{O(\log n)} E[X_i] = O(\log n)$.