This homework is due on gradescope Friday May 19th at 11:59pm. Remember to justify your work even if the problem does not explicitly say so. Writing your solutions in \LaTeX is recommend though not required. 

Recommended practice problems: Chapter 5, problems 15, 20, 26, 32.

**Question 1** (Optimal Caching Exchange Argument, 25 points). In our optimal caching problem, suppose that we have a cache of size $k = 3$. Suppose that the schedule given below is used to execute the given sequences of accesses. What are the intermediate schedules produced by the exchange argument discussed in class?

Accesses: ABCDCBACEDABED

---

Register 1: A E A E  
Register 2: B C D  
Register 3: CD B

**Question 2** (Rearrangement, 25 points). Given two lists $S$ and $T$ of $n$ numbers each, find a way to sort each of them $S = \{s_1, s_2, \ldots, s_n\}$ and $T = \{t_1, t_2, \ldots, t_n\}$ so that $\sum_{i=1}^n s_i t_i$ is as large as possible. For example, of $S = T = \{1, 2\}$, we could get either $1 \cdot 1 + 2 \cdot 2 = 5$ or $1 \cdot 2 + 2 \cdot 1 = 4$, and would prefer to find the former.

**Question 3** (Mining for Maximum Value, 25 points). Dirk is going to make money as a miner today. He has $N$ minutes of time that he can spend on this operation and several veins of ore that he can work on. The $i$th vein would take $t_i$ minutes to mine out completely, but the ore would be worth a total value of $v_i$. Of course if Dirk spends only a fraction $\alpha t_i$ time mining that ore (for some $0 \leq \alpha \leq 1$) he will get ore worth $\alpha v_i$. Give an efficient algorithm for finding a schedule for Dirk that gets him as much total value in ore as possible and prove that it is correct.

**Question 4** (Greedy Algorithms for Shortest Paths in DAGs, 25 points). Let $G$ be a DAG with vertex set $v_1, v_2, \ldots, v_n$ with edges going from $v_i$ to $v_j$ only for $j > i$. Suppose that the edges of $G$ are weighted with weights $\ell(v_i, v_j)$, and that we want to find the shortest path from $v_1$ to $v_n$. For each of the proposed greedy algorithms for this problem provide a counter-example. In particular, give an explicit graph $G$ and edge weights so that the greedy algorithm does not produce the optimal path. Show both the path produced by the greedy algorithm and the optimal one.

(a) Starting from $v_1$ build the path one vertex at a time. From vertex $v_i$, take the edge to $v_j$ with $j$ as large as possible. [5 points]

(b) Starting from $v_1$ build the path one vertex at a time. From vertex $v_i$, take the edge to $v_j$ with $\ell(v_i, v_j)$ as small as possible. [5 points]

(c) Starting from $v_1$ build the path one vertex at a time. From vertex $v_i$, take the edge to $v_j$ with $\ell(v_i, v_j)/(j - i)$ as small as possible. [5 points]

(d) Let $S$ be a set of vertices, initially $\{v_1\}$ and $T$ a set of edges initially 0. Repeatedly find the edge in $G$ from a vertex in $S$ to a vertex not in $S$ that has the smallest weight among such edges. Add this edge to $T$ and add the vertex on the other end of this edge to $S$. Repeat until $v_n$ has been added to $S$ and then take the unique path from $v_1$ to $v_n$ using only edges of $T$. [10 points]

**Question 5** (Extra credit, 1 point). Approximately how much time did you spend working on this homework?