CSCI 670:
Advanced Analysis of Algorithms Fall 2021

Quiz 1

Name:

Student ID:

Email Address:

Problem 1. *Consider an array of integers $a_1, \dots a_n$. We say a sequence π is a permutation of length n if π contains every element of $\{1, 2, \dots n\}$ exactly once and its length is n. Design an poly(n) algorithm to find a permutation π of length n such that ℓ_1 norm of $|\pi - a|_1$ is the smallest, i.e. $\sum_{i=1}^{n} |\pi_i - a_i|$.*

Solution 1. *For a given array a, let π_i^* be the index of a_i when the sequence a is sorted in ascending order. It is easy to see that π^* is a permutation. It remains to prove that π^* is the optimal permutation. To do so, we will prove that for any solution π and indices i, j if $a_i > a_j$ and $\pi_i < \pi_j$, swapping π_i and π_j weakly improves the result. Now, we consider the all possible orderings of a_i, a_j, π_i, π_j and prove the claim.*

- *$a_j < a_i < \pi_i < \pi_j$ (or $\pi_i < \pi_j < a_j < a_i$ by symmetry): swapping does not effect, see*

$$\pi_i - a_i + \pi_j - a_j = \pi_i - a_j + \pi_j - a_i.$$

- *$a_j < \pi_i < a_i < \pi_j$ (or $\pi_i < a_j < \pi_j < a_i$ by symmetry) swapping does improve, see*

$$a_i - \pi_i + \pi_j - a_j > \pi_i - a_j + \pi_j - a_i \qquad (a_i - \pi_i > \pi_i - a_i)$$

- *$a_j < \pi_i < \pi_j < a_i$ (or $\pi_i < a_j < a_i < \pi_j$ by symmetry) swapping does not effect, see*

$$a_i - \pi_i + \pi_j - a_j = \pi_i - a_j + a_i - \pi_j \qquad (\pi_j - \pi_i > \pi_i - \pi_j).$$

One can find such a permutation π^* by sorting the sequence a and a constant number of passes over a. So, we can obtain the optimal permutation to this problem in $O(n \log n)$.

Problem 2. *Let $G = (V, E)$ be an undirected graph. For every subset S of nodes of G, define*

$$f(S) = |\{v \mid v \in S \text{ or } v \text{ has a neighbor in } S\}|$$

as number of the nodes which are in S or have at least one neighbor in S. Prove that function $f : V \to N$ is submodular.

Solution 2. *Let $F(S) := \{v \mid v \in S \text{ or } v \text{ has a neighbor in } S\}$. Observe that $f(S) = |F(S)|$ for any subset S of vertices. Moreover for any $x \notin S$,*

$$f(S \cup \{x\}) - f(S) = |F(S \cup \{x\}) \setminus F(S)| = |F(\{x\}) \setminus F(S)| \tag{1}$$

where it follows by the fact that $F(S \cup \{x\}) = F(S) \cup F(\{x\})$.

For any $S \subseteq T \subseteq V$, we know that $F(S) \subseteq F(T)$, i.e. f is a monotone increasing set function. Moreover, observe that

$$\begin{aligned}
f(S \cup \{x\}) - f(S) &= |F(S \cup \{x\}) \setminus F(S)| \\
&= |F(\{x\}) \setminus F(S)| && (\textit{identity } (1)) \\
&\geq |F(\{x\}) \setminus F(T)| && (F(S) \subseteq F(T)) \\
&= |F(T \cup \{x\}) \setminus F(T)| \\
&= f(T \cup \{x\}) - f(T).
\end{aligned}$$

Therefore, f is a submodular function.

Problem 3 (Knapsack). *Suppose you are given a set of n items with costs c_i and values v_i. The knapsack problem is to find a set of items with maximum total value such that their total cost does not exceed the budget B. Without loss of generality, we can assume that all items meet the budget constraint, i.e. $c_i \leq B$.*

(a) Design an $O(n \log n)$ algorithm to find optimal solutions for the knapsack problem with divisible items. In other words, any fraction of an item can be in the knapsack.

(b) A feasible solution S is said to be 1/2-approximation to knapsack problem if

$$\sum_{i \in S} v_i \geq \frac{1}{2} \cdot OPT$$

where OPT is the total value of items in the optimal solution. Design an $O(n \log n)$ algorithm to knapsack problem that outputs a 1/2-approximate solution for any input.

Solution 3. *(a) Let's state the greedy algorithm for fractional knapsack problem.*

Algorithm 1 Greedy Fractional Knapsack Algorithm

```
Compute densities d_i := v_i / c_i for all i ∈ {1, ... n}
Sort items in decreasing order of d_i's. So, v_i and c_i are the value and the cost
of item i which are decreasing order of densities.
for i in {1, ... n} do
    Let C be the total cost of items in the current set of solution.
    if C + c_i ≤ B then
        Add item i to solution.
    else
        Add (B−C)/c_i fraction of item i to solution.
    end if
end for
```

One can show that replacing any fraction of an item in the solution with another item (possibly fractional) weakly decreases the total value. We left proof of this claim to the reader.

(b) Let OPT_f be the optimal value of the relaxed problem. Observe that a feasible solution to the 0-1 knapsack problem is feasible for the fractional knapsack problem, or we can say $OPT_f \geq OPT$.

Let's state an algorithm for 0-1 knapsack problem.

Algorithm 2 0-1 Knapsack Algorithm

Run the Algorithm 1.
Let S be the set of items belonging to solution and k be the index of the item fractionally included to the solution.
if $\sum_{i \in S} v_i \geq v_k$ **then**
  Output S
else
  Output k
end if

The following inequality proves that Algorithm 2 is $\frac{1}{2}$ approximation to 0-1 Knapsack problem.

$$2 \cdot \max \left\{ v_k, \sum_{i \in S} v_i \right\} \geq v_k + \sum_{i \in S} v_i \geq OPT_f \geq OPT.$$

Problem 4. *Let E be a finite set of elements, and $f : 2^E \to \mathbb{R}$ be a set function. We say that f is submodular if it satisfies one of the following equivalent conditions.*

1. $f(T \cup \{e\}) - f(T) \leq f(S \cup \{e\}) - f(S)$ *for all* $S \subseteq T \subseteq E$ *and* $e \in E \setminus T$.
2. $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$ *for any* $S, T \subseteq E$

Prove that two conditions are equivalent.

Solution 4.

- *(*$2 \implies 1$*): Let* $S \subseteq T \subseteq E$ *and* $e \in E \setminus T$. *Define* $S' = S \cup \{e\}, T' = T$ *and apply the second inequality.*

$$\begin{aligned}
&f(S') + f(T') \geq f(S' \cup T') + f(S' \cap T') \\
\implies &f(S' \cup T') - f(T') \leq f(S') - f(S' \cap T') \\
\implies &f(T \cup \{e\}) - f(T) \leq f(S \cup \{e\}) - f(S)
\end{aligned}$$

- *(*$1 \implies 2$*): Consider arbitrary* $S, T \subseteq E$. *Let* $\{e_1, \dots e_k\}$ *be an enumeration of* $S \setminus T$. *Define* $S_0 = S \cap T, T_0 = T, S_j := S_0 \cup \bigcup_{i=1}^{j} e_i$, *and* $T_j := T_0 \cup \bigcup_{i=1}^{j} e_j$. *Observe that* $S_k = S$ *and* $T_k = S \cup T$. *Now, let's apply the first inequality for* k *times:*

$$\begin{aligned}
&f(T_{i-1} \cup \{e_i\}) - f(T_{i-1}) \leq f(S_{i-1} \cup \{e_i\}) - f(S_{i-1}) \quad && \forall i \in 1, \dots k \\
\iff &f(T_i) - f(T_{i-1}) \leq f(S_i) - f(S_{i-1}) && \forall i \in 1, \dots k
\end{aligned}$$

Summation of these k *inequalities side by side completes the proof.*

$$\begin{aligned}
&f(T_k) - f(T_0) \leq f(S_k) - f(S_0) \\
\implies &f(T \cup S) - f(T) \leq f(S) - f(S \cap T) \\
\implies &f(T \cup S) + f(S \cap T) \leq f(S) + f(T).
\end{aligned}$$