Задача 1-1 (30 баллов). Рассмотрим алгоритмическую задачу с конечным множеством входов $\mathcal{I}$ и произвольное вероятностное распределение на нем. Рассмотрим также некоторое конечное семейство $\mathcal{A}$ детерминированных алгоритмов для ее решения и вероятностное распределение на нем. Обозначим через $f_A(I)$ время работы алгоритма A на входе I.

1. Докажите неравенство

$$\min_{A\in\mathcal{A}} E_I\left[f_A(I)\right] \leqslant \max_{I\in\mathcal{I}} E_A\left[f_A(I)\right],$$

где матожидания взяты относительно соответствующих распределений. Сформулируйте данное неравенство в терминах соотношения сложности в среднем (по входам) и рандомизированной сложности.

Решение. В силу дискретности распределений величин $I \in \mathcal{I}$ и $A \in \mathcal{A}$ получаем, что $E_I(f_A(I)) = \sum_{I\in\mathcal{I}} P(I)f_A(I)$, а также $E_A(f_A(I)) = \sum_{A\in\mathcal{A}} Q(A)f_A(I)$, с соответствующими таблицами распределения P и Q.

Значит, требуемое неравенство следует из очевидной цепочки сравнений:

$$\min_{A\in\mathcal{A}} E_I\left[f_A(I)\right] = \min_{A\in\mathcal{A}} \sum_{I\in\mathcal{I}} P(I)f_A(I) \leqslant \sum_{A\in\mathcal{A}}\sum_{I\in\mathcal{I}} P(I)Q(A)f_A(I) = E_{I,A}(f_A(I)),$$

а также

$$E_{I,A}(f_A(I)) = \sum_{I\in\mathcal{I}}\sum_{A\in\mathcal{A}} P(I)Q(A)f_A(I) \leqslant \max_{I\in\mathcal{I}} \sum_{A\in\mathcal{A}} Q(A)f_A(I) = \max_{I\in\mathcal{I}} E_A\left[f_A(I)\right].$$

Само же неравенство можно интерпретировать как то, что время, затрачиваемое лучшим в среднем (по входу) алгоритмом, всегда не больше, чем время, затрачиваемое на наихудший вход некоторого алгоритма "среднячка".

Это показывает связь между детерминированными алгоритмами с минимальной средней сложностью и рандомизированными алгоритмами с максимальной ожидаемой сложностью по всем входам.

2. Придумайте и сформулируйте определение *рандомизированного* алгоритма сортировки в модели решающих деревьев. Докажите в этой модели оценку $\Omega(n \log n)$ для сложности произвольного рандомизированного алгоритма сортировки, основанного на сравнении ключей.

Решение. Рандомизированность алгоритма сортировки в модели решающих деревьев можно интерпретировать как выбор случайных элементов сравнения за место детерминированных индексов элементов.

В ванильном детерминированном алгоритме сортировки в модели решающих деревьев из-за $n!$ количества перестановок и необходимости (для корректности) наличия соответствующих (как минимум) $n!$ листьев, получаем $\log_2 n!$ глубину дерева, которая асимптотически (по формуле Стирлинга) равна $n \log n$.

Рандомизированный алгоритм может давать разные деревья на одном и том же входе в зависимости от draw случайного числа, но ожидаемая глубина его выполнения всё равно ограничена снизу $\Omega(n \log n)$, поскольку даже при наличии случайности, алгоритм должен учитывать все возможные перестановки входов, что требует по меньшей мере $\log_2(n!)$ шагов.