Name: ____________________

Entry number: ____________________

- Always try to give algorithm with best possible running time. The points that you obtain will depend on the running time of your algorithm. For example, a student who gives an $O(n)$ algorithm will receive more points than a student who gives an $O(n^2)$ algorithm.
- You are required to give proofs of correctness whenever needed. For example, if you give a greedy algorithm for some problem, then you should also give a proof why this algorithm outputs optimal solution.
- **Use of unfair means will be severely penalized.**

There are 3 questions for a total of 40 points.

---

1. Prove the following two properties of the Huffman encoding scheme:

(5) (a) If some character occurs with frequency more than 2/5, then there is guaranteed to be a codeword of length 1.

(5) (b) If all characters occur with frequency less than 1/3, then there is guaranteed to be no codeword of length 1.

(10) 2. You are given an unsorted integer array A containing n distinct integers (assume n is a power of 2). You are supposed to design an algorithm that finds the minimum and the second minimum element in the array A. The running time for this problem is measured in terms of the number of pairwise comparisons that your algorithm performs in the worst case. (Note that you can find the minimum and the second minimum using $(2n - 3)$ comparisons by a linear scan.) Design a divide-and-conquer algorithm that solves the problem using at most $(3n/2 - 2)$ comparisons.

(**Extra Credit (5 points)**: *Design an algorithm that solves the problem using at most* $(n + \lceil \log n \rceil - 2)$ *comparisons.*)

(20) 3. There are n jobs that are supposed to be scheduled on a single machine. With each job i, there is an associated duration $t(i)$ that denotes the time that job i will take to execute on the machine. Each job i also has a given weight $w(i)$ that denotes the importance of this job. Given a schedule (an ordering of jobs to be executed on the machine), the completion time of job i is the time at which this job is finished by the machine. Let the completion time of job i be denoted by $C(i)$. Design an algorithm for giving a schedule that minimizes $\sum_{i=1}^{n} w(i) \cdot C(i)$. Give proof of correctness and discuss running time.