Name: ______________________ **Entry number:** ______________

COL351 Quiz 1 Duration: 1 hour

Read the following instructions before you begin writing.

1. Keep a pen, your identity card, and optionally a water bottle with you. Keep everything else away from you, at the place specified by the invigilators.
2. Do not detach sheets. Write your entry number and name on every page. (We will detach sheets prior to grading.)
3. Answer only in the designated space. Think before you use this space. No additional space will be provided for writing answers. Blank pages will be provided for rough work on demand, but they cannot be used for writing answers.
4. No clarifications will be given during the exams. If something is unclear or ambiguous, make reasonable assumptions and state them clearly. The instructor reserves the right to decide whether your assumptions were indeed reasonable.
5. You may use any result discussed in class or tutorials without proving it. Similarly, you can use any algorithm discussed in class or tutorials as a "black-box" i.e., without reproducing any details of how it works.

---

1. (8 points) The *median* of a real-valued (discrete) random variable X is defined to be any real number μ that satisfies $\Pr[X < \mu] \leq 1/2$ and $\Pr[X > \mu] \leq 1/2$. The distribution of such a random variable is given to you as an array $[(x_1, p_1), \ldots, (x_n, p_n)]$, where $p_1 = \Pr[X = x_i]$, $\sum_{i=1}^{n} p_i = 1$. Note that the x_i's need not be in a sorted order. Design an algorithm that, given such a distribution of a random variable X, returns its median in $O(n)$ time.

We use the algorithm for the n liquids problem from the tutorial. Treat the x_i's as ~~s~~ densities, p_i's as volumes of liquids mixed, and take V, the volume removed, to be $1/2$.

Find the volume of each liquid removed, and report the liquid with min density whose nonzero volume was removed as the median. Let that liquid be k.

Proof of correctness: We have

$$\sum_{i: x_i < x_k} p_i < \frac{1}{2} \leq \sum_{i: x_i \leq x_k} p_i$$

$\therefore \Pr[X < x_k] < \frac{1}{2}$ and $\Pr[X > x_k] = 1 - \sum_{i: x_i \leq x_k} p_i \leq \frac{1}{2}$.

Name: ______________________ **Entry number:** ________________

COL351 Quiz 1 Duration: 1 hour

2. (12 points) A set S of $2n$ points on the circumference of a circle with unit radius is given to you, and your job is to draw exactly n chords such that each of the given points lies on exactly one chord. While you might notice that the number of ways of doing this equals $(2n-1)!! = (2n-1) \times (2n-3) \times \cdots \times 3 \times 1$, a.k.a. the *double factorial* of $2n-1$, your job is to find a way that minimizes the total length of the chords.

   Assume that the circle is centered at $(0,0)$, and S is given as a sorted array $[\theta_1, \ldots, \theta_{2n}]$ of distinct numbers in the interval $[0, 2\pi)$, where the entry θ_i represents the point $p_i = (\cos(\theta_i), \sin(\theta_i))$. Design a polynomial-time algorithm that takes such an array of points as input, and outputs the minimum possible total length of n chords such that each of the given points lies on exactly one chord. Provide short proofs of correctness and polynomial running time.

   You are allowed to use the following results without proving them.

   1. The sum of the lengths of diagonals of any convex quadrilateral is no less than the sum of the lengths of any pair of its opposite sides.
   2. If a chord of a unit radius circle makes angle θ at its center, then the length of the chord is $|2\sin(\theta/2)|$. Thus, the distance between points p_i and p_j is $d_{i,j} = |2\sin((\theta_i - \theta_j)/2)|$.

   You can also assume that $d_{i,j}$ can be computed exactly in constant time.

The first result implies $\exists$ an optimal solution in which no two of the n chords intersect.

$\therefore$ $\exists$ an optimal solution in which

① p_1 is paired with p_j for some even j

② $p_2, \ldots, p_{j-1}$ are paired among themselves optimally.

③ $p_{j+1} \cdots p_{2n}$ are paired among themselves optimally.

We design a DP algorithm with a 2D table M, whose rows and columns are indexed by $\{1 \cdots 2n\}$.

For all i, j such that $i < j$ and i, j have opposite parity, we store in $M[i,j]$ the opt cost of pairing points $p_i \cdots p_j$ among themselves.

$M[i,j]$ is computed as follows.

$$M[i,j] = \min_{\substack{k \in \{i+1, \ldots, j\} \\ \text{s.t. } k \text{ has opposite} \\ \text{parity as } i}} d_{i,k} + M[i+1, k-1] + M[k+1, j]$$

Base case: $M[i,j] = 0$ if $i > j$. Final ans: $M[1, 2n]$.

Computation order: Increasing order of $j-i$; entries with the same value of $j-i$ can be computed in an arbitrary order.

Running time = Table size $\times$ time per entry

$= O(n^2) \times O(n)$

$= O(n^3) \rightarrow$ polynomial in n.