1. [20 points] PICK ONE *No justification is needed for this problem.*

(a) Which one of the following is incorrect for Two Dimensional Range Trees?

   i. Counting range queries take $O(\log^2(n))$ time.
   ii. Reporting range queries take $O(\log^2(n) + k)$ time.
   iii. Construction time is $\Theta(n \log n)$.
   iv. Space complexity is $\Theta(n^2 \log n)$.

(b) CLIQUE-DEGREE-4 problem: Given a graph $G = (V, E)$ where all vertices have degree at most 4, and an integer k, determine whether V has a subset S of size at least k that forms a clique in G. Which of the following statement(s) is/are correct?
(a) CLIQUE-DEGREE-4 is NP-Hard; (b) CLIQUE-DEGREE-4 is in P; (c) Approximating CLIQUE-DEGREE-4 is NP-Hard; (d) There exists a polynomial time reduction from CLIQUE-DEGREE-4 to VERTEXCOVER problem.

   i. (a) and (d)
   ii. (b) and (d)
   iii. (a) and (c)
   iv. (b) and (c)
   v. All of them

(c) MAJORITY-CHECK problem: Given a set S of n real numbers, determine whether more than half the numbers in S are exactly the same. Assuming two numbers can be compared in constant time, which of the following statement(s) is/are correct?
(a) MAJORITY-CHECK is in NP; (b) There exists no comparison-based algorithm solving MAJORITYCHECK in $\Theta(n)$ time; (c) MAJORITY-CHECK is proven to be NP-hard via a reduction from SUBSETSUM problem.

   i. only (a)
   ii. (a) and (b)
   iii. (a) and (c)
   iv. All of them
   v. None of them

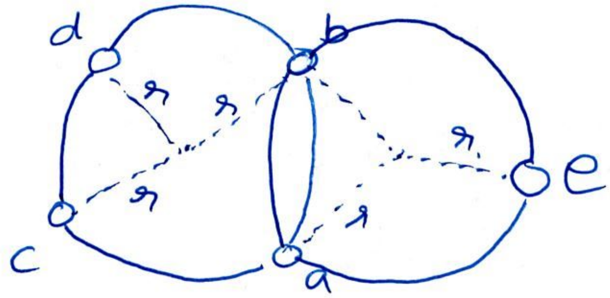
(d) MAX-SAT: Given a CNF (Conjunctive Normal Form) Boolean formula and a positive integer k, does there exist a truth assignment that satisfies at least k clauses.
Which one of the following is correct about this problem?

   i. This problem can be solved in polynomial time since k is a small number.
   ii. This problem is NP-hard since it is a special case of the SAT problem.
   iii. This problem is NP-hard since it is a generalization of the SAT problem.
   iv. This problem can be shown to be NP-hard by giving a reduction from MAX-SAT to CLIQUE problem.
   v. None of the above

2. [30 points] GEOMETRIC ALGORITHMS - CO-CIRCULARITY CHECK

Following procedures each with planar input points and constant running time are provided: CCW(p, q, r) decides if r is LEFTOF, ON, or RIGHTOF the oriented line going through p, q. INCIRCLE(p, q, r, s) decides whether s is INSIDE, ON, or OUTSIDE the unique circle going through p, q, r.

(a) **Draw** five points a, b, c, d, e such that the radii of the circles going through triples of points (a, b, c), (a, b, d), and (a, b, e) are all the same, a, b, c, d are co-circular, and yet a, b, c, e are not co-circular, i.e., circumradius(a, b, c) = circumradius(a, b, d) = circumradius(a, b, e), INCIRCLE (a, b, c, d) = ON, and INCIRCLE (a, b, c, e) ≠ ON.



(b) Given n points in the plane (no three of which are co-linear) we want to determine whether any four of them are co-circular. Naive algorithm would take $O(n^4)$ time. **Design** and **analyze** an algorithm that solves this problem in $O(n^3 \log n)$ time.
(*Hints: Sorting takes* $O(n \log n)$ *time. Recall the co-linearity checking algorithm.*)

Algorithm:

For every pair of points p, q in the input set of point P

1. Sort all other points r in P with respect to circumradius of pqr using the INCIRCLE test.
[Separate the points into 2 lists ⓐ one of the points in the left of the directed line (or) oriented line pq ⓑ one of the points in the right of the directed line pq (use CCW)].

2. Check neighbour elements r_1 and r_2 in the lists to see if p, q, r_1, r_2 are co-circular using INCIRCLE test (do this for both the list.

Running time: $\underset{\substack{\uparrow \\ \text{picking pairs}}}{({}^nC_2)} \times \underset{\substack{\uparrow \\ \text{sorting}}}{n \log n} = O(n^3 \log n)$ time

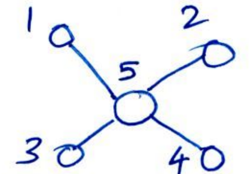
3. [30 points] NP-COMPLETENESS - STRONGLY INDEPENDENT SETS ↙IS

Consider a simple graph $G = (V, E)$. A subset $S_1 \subseteq V$ is called an *independent set*, if no two vertices in S_1 have an edge (path of length one) between them. A subset $S_2 \subseteq V$ is called a SIS→ *strongly independent set*, if no two vertices in S_2 have a path of length one or two between them, i.e., $\forall\, u, v \in S_2$, we have $(u, v) \notin E$, and $\nexists\, w \in V$ such that $(u, w) \in E$ and $(w, v) \in E$.

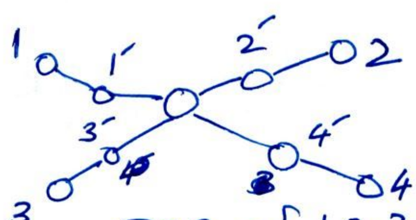
INDEPENDENTSET PROBLEM: Given a simple graph G and an integer k, determine if G contains an independent set of size at least k.

STRONGLYINDEPENDENTSET PROBLEM: Given a simple graph G and an integer k, determine if G contains a strongly independent set of size at least k.

(a) **Draw** a connected graph that has an independent set of size 4. **Draw** another connected graph that has a strongly independent set of size 4.



IS = {1,2,3,4}



SIS = {1,2,3,4}

(b) **Prove** that STRONGLYINDEPENDENTSET PROBLEM is NP-Complete. (*Hints:* INDEPENDENTSET PROBLEM *is NP-hard. For your reduction, consider additional vertices positioned strategically and forming a clique among themselves.*)

To prove that the problem SIS is NP-COMPLETE

Step1 : SIS ∈ NP

Let $S \subset V$,

i, We can verify that $|S| \leq k$

ii, For every $u, v \in S$
verify if Distance between 'u' and 'v' is atleast 2

This can be done in polynomial time

$\Rightarrow$ SIS ∈ NP — 1

Step2: SIS ∈ NP-HARD

we generate an instance of SIS from IS

$IS \le^P SIS$ → IS to SIS Reduction in polynomial time.

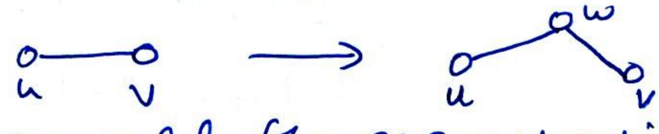
[by Hint] → (Known to be NP-~~COMPLETE~~ HARD)

Let $\langle G, K \rangle$ be the input instance of IS.

we map/correspond it to the input instance of SIS.

Reduction: Let 'e' be an edge in G, $(G = V, E)$, $e = (u, v)$,

i, Introduce a new vertex 'w' by replacing the edge as $(u,w), (w,v)$.



ii, we form clique over all the new vertices w and call this new graph $G' = (V', E')$

$V' = V \cup V_w$ where V_w is set of all the new 'w''s

E' = Edges formed as discussed above.

Proof on Reduction: (we have to prove in both the directions)

i) ⇒ Let $S \subset V$ be IS in G then we show that same set of vertices form SIS in G'.

By our construction if $d(u,v) > 1$ in G then $d(u,v) > 2$ in G'.

⇒ S has pairwise distance of atleast 3 in G'. and $S \subset V \subset V'$ in G' with $|S| \le K$.

Thus, we say the claim.

ii, ⇐ Let $S' \subset V'$ be SIS in G' then we show that same set of vertices form IS in G.

S' cannot contain any of the new vertices 'w' because any other vertex in G' is reachable from 'w' in path of length '2'. ⇒ $S' \subset V'$ and $S' \subset V$.

S' is thus an IS in G' and $|S'| \le K$.

$\Rightarrow$ SIS is reduced from IS

ie, $IS \leq^{P} SIS$.

$\Rightarrow$ SIS is atleast as hard as IS

and IS $\in$ ~~NP COMPLETE~~ NP-HARD (by hint)

$\Rightarrow$ SIS $\in$ NP-HARD — (2)

Step 3: SIS $\in$ NP (by (1))

SIS $\in$ NP-HARD (by (2))

Since SIS is NP and NP-HARD we can say that SIS $\in$ NP-COMPLETE.

4. [30 points] APPROXIMATION - PACK YOUR BOOKS, YOU GOT A JOB AT GOOGLE
Imagine that you just graduated UF and got a job at Google. For your move to California, you need to pack all your books to boxes each of which has capacity a real number between 1 and 2. For simplicity, assume that each book has weight a real number between 0 and 1. You want to minimize the number of boxes that you use. More formally:

PACKBOOKS PROBLEM: Given n books with weights $w_1, \ldots, w_n$ where $\forall$ i, $0 < w_i < 1$, m boxes with capacities $c_1, \ldots, c_m$ where $\forall$ j, $1 < c_j < 2$, and an integer k, is it possible to pack all books in at most k boxes such that total weight in each box is within its capacity.

(a) **Prove** that PACKBOOKS PROBLEM is NP-hard. (*Hints: Consider a special case of this problem that you are familiar with. No reduction necessary.*)

BIN PACKING problem is a special case of PACKBOOKS problem.
when all boxes have capacity eaual to 1. $[C_j = 1]$.
Since we know that BIN PACKING $\in$ NP-HARD
we can say that PACKBOOKS $\in$ NP-HARD.

(b) **Design** and **analyze** an approximation algorithm for the minimization version of the PACKBOOKS PROBLEM with an approximation ratio at most 4.

Algorithm: (First Fit)
For every book b:
Place b in the First box
which fits b.

The above algorithm gives a 4-Approximation
Reason: (a) Each box has a capacity to hold books
upto a weight of 2
$\Rightarrow$ Lower bound is $\frac{\sum_{i=1}^{n} w_i}{2}$.

(b) Atmost one box is less than half full.
Let C_1 and C_2 be capacities of 2 boxes. we will not place books in box 2 until box 1 ~~is~~ cannot hold the book
If both are half full then books in box 2 can be placed in box 1 ~~box~~ as box 1 appears before box 2.

4-approximation proof

Let 'N' be the total no: of boxes used by our algorithm. Let OPT represent the optimal solution

$$\frac{1}{2}(N-1) \leq \sum_{i=1}^{n} w_i \leq 2 \times (OPT).$$

$$\Rightarrow \quad N \leq (4 \times (OPT)) + 1$$

$\Rightarrow$ It is a 4-approximation ratio algorithm.