



Jahangirnagar University
Department of Computer Science and Engineering
3rd Year 1st Semester B.Sc. (Hons.) Tutorial Exam -2022

Course Title: Computational Geometry
Time: 80 minutes

Course No: CSE-305
Full Marks: 30+30

[দেখি সব পারি কিনা]

Part-I

- Q1. a) Upon the circumference of the inner circle with radius r are placed n outer circles each with radius R in such a way that two consecutive outer circles touch each other and the inner circle touches all the outer circles just as shown in the diagram below:



- i. Formulate an equation to compute R given the value of r and n . [absolute error or relative error should not exceed 10^{-5}] [5]
- ii. If $n = 100$, $r = 100$ then $R = ?$ [5]

- b) Given three non-collinear points $A(x_1, y_1)$, $B(x_2, y_2)$ and $C(x_3, y_3)$ on a 2D plane. Write an algorithm to find the minimum distance of the line segment specified by $P(x, y)$ and $Q(x, y)$ from the center, S of the circle passing through A , B and C . [6]

- c) Write an algorithm to calculate the number of lattice points on a line segment with ends points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$. [4]

- d) Given N points as $P_i (x_i, y_i); 0 \leq i < N$ on a 2D space and a specific point $Q (x, y)$. Write an algorithm of at most **log linear** ($N \log N$) time complexity to find the minimum number of lines going through Q in order to cover all the points. [10]

Part-II

Q2.

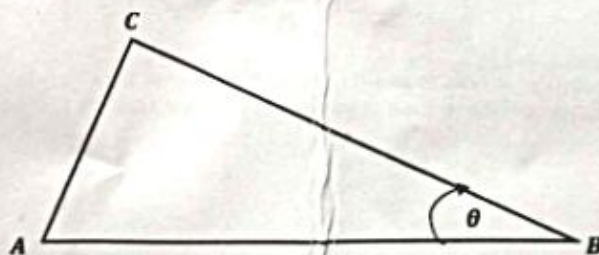
- a) Given two points as $A (10, 20)$ and $B (30, 10)$. Now answer the following:
- Represent the line AB in parametric form considering t as the parameter. [2]
 - Analyze the characteristic of t . [2.5]

- iii. Find a unit vector which will be perpendicular to \overrightarrow{AB} .

[2.5]

- b) Given two end points of a line segments as $M(0,0)$ and $N(2,0)$. Find the minimum distance of the line segment from the given point $P(4,0)$. [5]

- c) A triangle has to be drawn in such a way that the length of the largest side is given as $AB = x$ and the sum of the other two sides is given as $AC + BC = S$ with a constraint that BC maintains an angle θ with AB as shown in the picture. [10]



- i. Analyze the case when the values of AC and BC cannot be determined if any.

- ii. Write an algorithm to find the value of AC and BC . [absolute error or relative error should not exceed 10^{-6}]

- iii. Find the value of AC and BC given that $x = 12$, $S = 18$, $\theta = 90^\circ$. [only up to 3rd iteration if iterations are required]

- d) Two line-segments are defined as $P_1(1,2)$, $Q_1(1,7)$ and $P_2(5,5)$, $Q_2(-5,5)$. Demonstrate the [8]
execution trace of the algorithm to check if they intersect.

- b) Let $P = (P_0, P_1, P_2, \dots, P_{n-1})$ be a sequence of n points where $(P.x, P.y) \in \mathbb{Z}$ on a $2D$ plane. [6]
Someone claims that the following algorithm can identify whether or not P describes the boundary of a convex polygons in clockwise order.

```
bool is_convex( $p_0, \dots, p_{n-1}$ ) {  
    for  $i = 0, \dots, n-1$ :  
        if ( $p_i, p_{(i+1) \bmod n}, p_{(i+2) \bmod n}$ ) form a rightturn:  
            return false;  
    return true;  
}
```

Disprove the claim and describe a correct algorithm to solve the problem.


```

    if (x == 0) return true;
    if (x < 0) return false;
    if (x > 0) return true;
    return false;
}

```

3. ... (b) ... (c) ... (d) ... (e) ... (f) ... (g) ... (h) ... (i) ... (j) ... (k) ... (l) ... (m) ... (n) ... (o) ... (p) ... (q) ... (r) ... (s) ... (t) ... (u) ... (v) ... (w) ... (x) ... (y) ... (z) ... (aa) ... (ab) ... (ac) ... (ad) ... (ae) ... (af) ... (ag) ... (ah) ... (ai) ... (aj) ... (ak) ... (al) ... (am) ... (an) ... (ao) ... (ap) ... (aq) ... (ar) ... (as) ... (at) ... (au) ... (av) ... (aw) ... (ax) ... (ay) ... (az) ... (ba) ... (bb) ... (bc) ... (bd) ... (be) ... (bf) ... (bg) ... (bh) ... (bi) ... (bj) ... (bk) ... (bl) ... (bm) ... (bn) ... (bo) ... (bp) ... (bq) ... (br) ... (bs) ... (bt) ... (bu) ... (bv) ... (bw) ... (bx) ... (by) ... (bz) ... (ca) ... (cb) ... (cc) ... (cd) ... (ce) ... (cf) ... (cg) ... (ch) ... (ci) ... (cj) ... (ck) ... (cl) ... (cm) ... (cn) ... (co) ... (cp) ... (cq) ... (cr) ... (cs) ... (ct) ... (cu) ... (cv) ... (cw) ... (cx) ... (cy) ... (cz) ... (da) ... (db) ... (dc) ... (dd) ... (de) ... (df) ... (dg) ... (dh) ... (di) ... (dj) ... (dk) ... (dl) ... (dm) ... (dn) ... (do) ... (dp) ... (dq) ... (dr) ... (ds) ... (dt) ... (du) ... (dv) ... (dw) ... (dx) ... (dy) ... (dz) ... (ea) ... (eb) ... (ec) ... (ed) ... (ee) ... (ef) ... (eg) ... (eh) ... (ei) ... (ej) ... (ek) ... (el) ... (em) ... (en) ... (eo) ... (ep) ... (eq) ... (er) ... (es) ... (et) ... (eu) ... (ev) ... (ew) ... (ex) ... (ey) ... (ez) ... (fa) ... (fb) ... (fc) ... (fd) ... (fe) ... (ff) ... (fg) ... (fh) ... (fi) ... (fj) ... (fk) ... (fl) ... (fm) ... (fn) ... (fo) ... (fp) ... (fq) ... (fr) ... (fs) ... (ft) ... (fu) ... (fv) ... (fw) ... (fx) ... (fy) ... (fz) ... (ga) ... (gb) ... (gc) ... (gd) ... (ge) ... (gf) ... (gg) ... (gh) ... (gi) ... (gj) ... (gk) ... (gl) ... (gm) ... (gn) ... (go) ... (gp) ... (gq) ... (gr) ... (gs) ... (gt) ... (gu) ... (gv) ... (gw) ... (gx) ... (gy) ... (gz) ... (ha) ... (hb) ... (hc) ... (hd) ... (he) ... (hf) ... (hg) ... (hh) ... (hi) ... (hj) ... (hk) ... (hl) ... (hm) ... (hn) ... (ho) ... (hp) ... (hq) ... (hr) ... (hs) ... (ht) ... (hu) ... (hv) ... (hw) ... (hx) ... (hy) ... (hz) ... (ia) ... (ib) ... (ic) ... (id) ... (ie) ... (if) ... (ig) ... (ih) ... (ii) ... (ij) ... (ik) ... (il) ... (im) ... (in) ... (io) ... (ip) ... (iq) ... (ir) ... (is) ... (it) ... (iu) ... (iv) ... (iw) ... (ix) ... (iy) ... (iz) ... (ja) ... (jb) ... (jc) ... (jd) ... (je) ... (jf) ... (jg) ... (jh) ... (ji) ... (jj) ... (jk) ... (jl) ... (jm) ... (jn) ... (jo) ... (jp) ... (jq) ... (jr) ... (js) ... (jt) ... (ju) ... (jv) ... (jw) ... (jx) ... (jy) ... (jz) ... (ka) ... (kb) ... (kc) ... (kd) ... (ke) ... (kf) ... (kg) ... (kh) ... (ki) ... (kj) ... (kk) ... (kl) ... (km) ... (kn) ... (ko) ... (kp) ... (kq) ... (kr) ... (ks) ... (kt) ... (ku) ... (kv) ... (kw) ... (kx) ... (ky) ... (kz) ... (la) ... (lb) ... (lc) ... (ld) ... (le) ... (lf) ... (lg) ... (lh) ... (li) ... (lj) ... (lk) ... (ll) ... (lm) ... (ln) ... (lo) ... (lp) ... (lq) ... (lr) ... (ls) ... (lt) ... (lu) ... (lv) ... (lw) ... (lx) ... (ly) ... (lz) ... (ma) ... (mb) ... (mc) ... (md) ... (me) ... (mf) ... (mg) ... (mh) ... (mi) ... (mj) ... (mk) ... (ml) ... (mm) ... (mn) ... (mo) ... (mp) ... (mq) ... (mr) ... (ms) ... (mt) ... (mu) ... (mv) ... (mw) ... (mx) ... (my) ... (mz) ... (na) ... (nb) ... (nc) ... (nd) ... (ne) ... (nf) ... (ng) ... (nh) ... (ni) ... (nj) ... (nk) ... (nl) ... (nm) ... (nn) ... (no) ... (np) ... (nq) ... (nr) ... (ns) ... (nt) ... (nu) ... (nv) ... (nw) ... (nx) ... (ny) ... (nz) ... (oa) ... (ob) ... (oc) ... (od) ... (oe) ... (of) ... (og) ... (oh) ... (oi) ... (oj) ... (ok) ... (ol) ... (om) ... (on) ... (oo) ... (op) ... (oq) ... (or) ... (os) ... (ot) ... (ou) ... (ov) ... (ow) ... (ox) ... (oy) ... (oz) ... (pa) ... (pb) ... (pc) ... (pd) ... (pe) ... (pf) ... (pg) ... (ph) ... (pi) ... (pj) ... (pk) ... (pl) ... (pm) ... (pn) ... (po) ... (pp) ... (pq) ... (pr) ... (ps) ... (pt) ... (pu) ... (pv) ... (pw) ... (px) ... (py) ... (pz) ... (qa) ... (qb) ... (qc) ... (qd) ... (qe) ... (qf) ... (qg) ... (qh) ... (qi) ... (qj) ... (qk) ... (ql) ... (qm) ... (qn) ... (qo) ... (qp) ... (qq) ... (qr) ... (qs) ... (qt) ... (qu) ... (qv) ... (qw) ... (qx) ... (qy) ... (qz) ... (ra) ... (rb) ... (rc) ... (rd) ... (re) ... (rf) ... (rg) ... (rh) ... (ri) ... (rj) ... (rk) ... (rl) ... (rm) ... (rn) ... (ro) ... (rp) ... (rq) ... (rr) ... (rs) ... (rt) ... (ru) ... (rv) ... (rw) ... (rx) ... (ry) ... (rz) ... (sa) ... (sb) ... (sc) ... (sd) ... (se) ... (sf) ... (sg) ... (sh) ... (si) ... (sj) ... (sk) ... (sl) ... (sm) ... (sn) ... (so) ... (sp) ... (sq) ... (sr) ... (ss) ... (st) ... (su) ... (sv) ... (sw) ... (sx) ... (sy) ... (sz) ... (ta) ... (tb) ... (tc) ... (td) ... (te) ... (tf) ... (tg) ... (th) ... (ti) ... (tj) ... (tk) ... (tl) ... (tm) ... (tn) ... (to) ... (tp) ... (tq) ... (tr) ... (ts) ... (tt) ... (tu) ... (tv) ... (tw) ... (tx) ... (ty) ... (tz) ... (ua) ... (ub) ... (uc) ... (ud) ... (ue) ... (uf) ... (ug) ... (uh) ... (ui) ... (uj) ... (uk) ... (ul) ... (um) ... (un) ... (uo) ... (up) ... (uq) ... (ur) ... (us) ... (ut) ... (uu) ... (uv) ... (uw) ... (ux) ... (uy) ... (uz) ... (va) ... (vb) ... (vc) ... (vd) ... (ve) ... (vf) ... (vg) ... (vh) ... (vi) ... (vj) ... (vk) ... (vl) ... (vm) ... (vn) ... (vo) ... (vp) ... (vq) ... (vr) ... (vs) ... (vt) ... (vu) ... (vv) ... (vw) ... (vx) ... (vy) ... (vz) ... (wa) ... (wb) ... (wc) ... (wd) ... (we) ... (wf) ... (wg) ... (wh) ... (wi) ... (wj) ... (wk) ... (wl) ... (wm) ... (wn) ... (wo) ... (wp) ... (wq) ... (wr) ... (ws) ... (wt) ... (wu) ... (wv) ... (ww) ... (wx) ... (wy) ... (wz) ... (xa) ... (xb) ... (xc) ... (xd) ... (xe) ... (xf) ... (xg) ... (xh) ... (xi) ... (xj) ... (xk) ... (xl) ... (xm) ... (xn) ... (xo) ... (xp) ... (xq) ... (xr) ... (xs) ... (xt) ... (xu) ... (xv) ... (xw) ... (xx) ... (xy) ... (xz) ... (ya) ... (yb) ... (yc) ... (yd) ... (ye) ... (yf) ... (yg) ... (yh) ... (yi) ... (yj) ... (yk) ... (yl) ... (ym) ... (yn) ... (yo) ... (yp) ... (yq) ... (yr) ... (ys) ... (yt) ... (yu) ... (yv) ... (yw) ... (yx) ... (yy) ... (yz) ... (za) ... (zb) ... (zc) ... (zd) ... (ze) ... (zf) ... (zg) ... (zh) ... (zi) ... (zj) ... (zk) ... (zl) ... (zm) ... (zn) ... (zo) ... (zp) ... (zq) ... (zr) ... (zs) ... (zt) ... (zu) ... (zv) ... (zw) ... (zx) ... (zy) ... (zz) ...

- e) Let $P \in \mathbb{Z}^2$ be a convex polygon, given as an array $P[0], P[1], \dots, P[n-1]$ of its $N (\leq 10^6)$ [8] vertices and $M (\leq 10^6)$ queries are provided. Each query contains a point $Q \in \mathbb{Z}^2$. Construct an algorithm of not more than **log linear time complexity** to find if Q lies inside, outside or on the boundary of the polygon.

- d) Let $P \subset \mathbb{Z}^2$ be a **convex** polygon, defined with the following vertices.

| V_0 | V_1 | V_2 | V_3 | V_4 | V_5 | V_6 | V_7 |
|--------|---------|---------|---------|---------|--------|--------|--------|
| (7, 1) | (13, 2) | (14, 7) | (10, 9) | (6, 10) | (5, 9) | (4, 6) | (4, 3) |

Determine the number of lattice points inside the convex polygon



Jahangirnagar University
Department of Computer Science and Engineering
3rd Year 1st Semester B.Sc. (Hons.) Tutorial Exam -2022

Course Title: Computational Geometry
Time: minutes

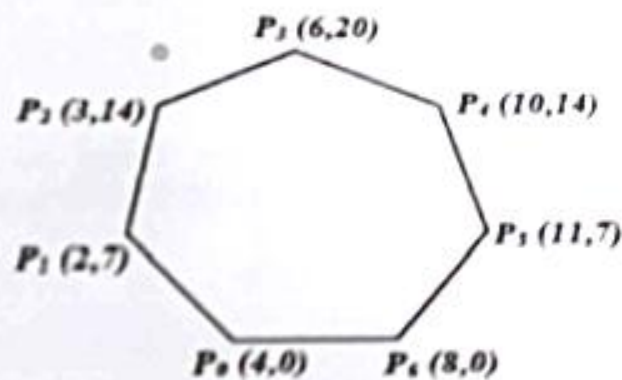
Course No: CSE-305
Full Marks: 30

। দেখি সব পারি কিনা

I-III

- Q1. a) Determine the upper tangent from R to the convex polygon P in Z^2 with the condition that:
 $R.x < P_i.x; 0 \leq i < n$ using an algorithm of $\log_2 n$ running time complexity.

●
 $R(-10, 8)$

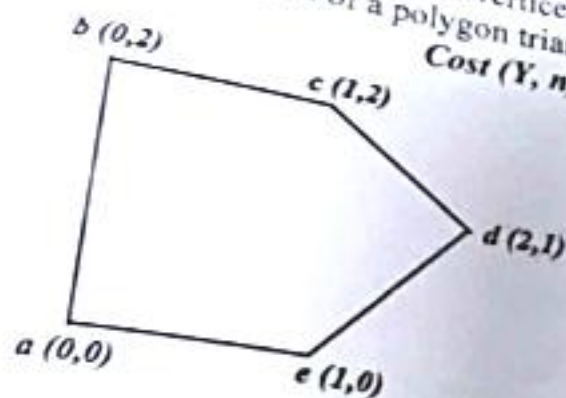




Department of Computer Science and Engineering
Jahangirnagar University

- b) A convex polygon Y of n vertices can be triangulated into $n-2$ triangles with respect to any vertex. The cost of a polygon triangulation is defined as: [10]

$$\text{Cost}(Y, n) = \Delta_1 + \Delta_2 + \dots + \Delta_i + \dots + \Delta_{n-2}$$

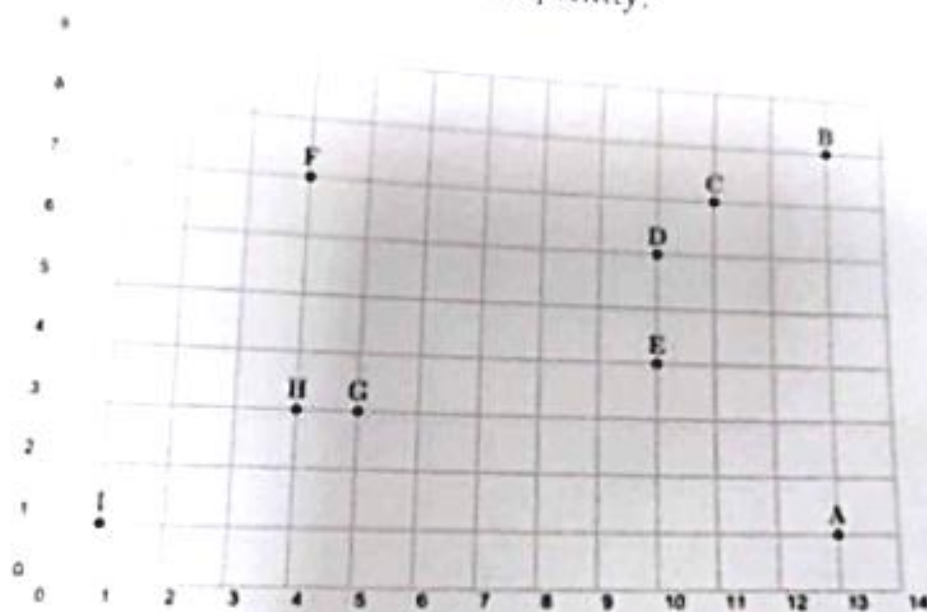


$$\Delta_i(a, b, c) = \text{dist}(a, b) + \text{dist}(b, c) + \text{dist}(a, c)$$

$$\text{dist}(p, q) = \sqrt{(p.x - q.x)^2 + (p.y - q.y)^2}$$

Determine the minimum cost of polygon triangulation considering the polygon to the left. Mention the time complexity of your algorithm.

- c) Construct a convex hull of the following set of points given in the diagram using an algorithm of at most log linear ($N \log_2 N$) running time complexity. [10]



If there are 10^6 points in the plane, could Jarvis March algorithm be used to solve this problem. Mention the case for which Jarvis March runs in $O(n)$ time complexity.

C. Geometry

①

- ✖ 3) Write an algorithm of $O(\log n)$ time complexity in order to inspect whether [10]
a given point lies inside a convex polygon or not.