Coding project: Intersection graphs

Group #2

Grader: Allouzi, Maha A.

Members:

1. Breitsch, Nathan W.
2. Chalasani, Tarun
3. Challa, Rohitkumar Reddy
4. Dharavath, Bharathi
5. Inti, Suchitra Ganga Bhavani Anusha
6. Kakumanu, Gayatri
7. Kaparthi, Rakesh
8. Kasetty, Santosh K

Programs:

1. Permutation graphs (**Breitsch, Nathan W**.)
   Input: Interactively input a number of vertices "n" and a permutation of numbers 1,2,…,n.
   Output: Draw an intersection model (two parallel lines; on one line numbers 1,2,…,n and on the other the permutation of them; points on two lines having the same number are connected by a segment) and the permutation graph obtained from the model (the intersection graph of those segments).

2. Triangle graphs (= degenerate trapezoid graphs) (**Chalasani, Tarun**)
   Input: Interactively input a number of vertices "n", a permutation of numbers 1,2,…,n and n intervals (for each interval ask to input start and finish points).
   Output: Draw an intersection model (two parallel lines; on one line the permutation of numbers 1,2,…,n and on the other intervals numbered from 1 to n. Ends of each interval i connect by segments with the point i on the other line; you get n triangles squeezed between two parallel lines) and the triangle graph obtained from the model (the intersection graph of those triangles).

3. Circular-arc graphs (**Challa, Rohitkumar Reddy**)
   Input: Interactively input a number of vertices "n" and a set of n circular arcs (for each arc ask to input start and finish points on the circle in degrees between 0 and 360; draw each circular arc from start to finish points in counterclockwise manner).
   Output: Draw an intersection model (a circle and n arcs on that circle) and the circular-arc graph obtained from the model (the intersection graph of those arcs).

4. Circle graphs (**Dharavath, Bharathi**)
   Input: Interactively input a number of vertices “n” and a set of n chord ends (for each chord ask to input start and finish points on the circle in degrees between 0 and 360).
   Output: Draw an intersection model (a circle and n chords of that circle) and the circle graph obtained from the model (the intersection graph of those chords).

5. 4-Polygon graphs (**Inti, Suchitra Ganga Bhavani Anusha**)
   Input: Interactively input a number of vertices “n”, a rectangle with n points on each side, and a set of n chord ends (for each chord ask to input start point on one side of the rectangle and finish point on another side of the rectangle).
   Output: Draw an intersection model (a rectangle and n chords of that rectangle) and the 4-polygon graph obtained from the model (the intersection graph of those chords).

6. 3-Polygon graphs (**Kakumanu, Gayatri**)
   Input: Interactively input a number of vertices “n”, a triangle with n points on each side, and a set of n chord ends (for each chord ask to input start point on one side of the triangle and finish point on another side of the triangle).
   Output: Draw an intersection model (a triangle and n chords of that triangle) and the 3-polygon graph obtained from the model (the intersection graph of those chords).

7. 2-boxicity graphs (**Kaparthi, Rakesh**)
   Input: Interactively input a number of vertices “n” and a set of n rectangles (sides parallel to axes; for each rectangle ask coordinates of the lower-left corner and of the upper-right corner).
   Output: Draw an intersection model (set of rectangles on the plane) and the 2-boxicity graph obtained from the model (the intersection graph of those rectangles).

8. Unit-disk graphs (**Kasetty, Santosh K**)
   Input: Interactively input a number of vertices “n”, a radius “r”, and a set of n disks of radius r each (for each disk ask the coordinates of its center).
   Output: Draw an intersection model (set of disks of radius r on the plane) and the unit-disk graph obtained from the model (the intersection graph of those disks).