Circle / Triangle / Rectangle / Square / Intersections / Overlaps / Point-in-Something
---
190 *** CIRCUMCENTER
191 *** lineSegment solid Rectangle intersection
378 ** line line intersection
438 * EASIER than 190 -- only CIRCUMFERENCE
453 *** circle circle intersection
460 *   rectangle rectangle overlap
476 *
477 *
478 **
10175 ** formula needed or do integration to find the formula.
10522 **
10573 *
10709 ****
10725 ***
10979 ***** intersections and Floyd Warshall (Graph)

Vector2D / Vector3D
---
190 ***
191 ***
378 **
453 ***
10242 *
10674 **** Tangents of two circles... 5 cases
10709 **** Better to Use VECTOR here!
11580 ***** (3D) http://uva.onlinejudge.org/external/115/11580.html Although Hard, it is worth noticing its figure. It might help understanding latitude, longitude.
[Timus] 1697 ***** (3D)
[Timus] 1703 **
[Timus] 1710 *
[TJU] 3114 *** (3D)

Packing Problems

10283 **
10286 *
10287 *** +BS
10289 **** +BS
10353 **** +BS
10402
10481 **** +BS

11009 **** BS could have been useful, but will get TLE

Binary Search (BS) [Bisection Method]

10287 **
10289 ***
10322 ****
10341 *
10345 ***
10353 ****
10372 **
10386 ***
10398 **
10481 ****
10566 **
10631 ***** Normals of Ellipse... 2 cases
10668 ** [Discussed in class]
10695 **** MANY CASES!!!

Geodesic Distance
---
http://en.wikipedia.org/wiki/Great-circle_distance
10517 ***
10598 ***
10809 ***** - Geodesic distance / Solving Using Parameter / Great Circle's Clear Concept.

Convex Hull -- Learn Jarvis March O(nh) and Graham Scan O(nlogn) Algorithm
---
http://www.cs.ucf.edu/courses/cot5520/GScanJMarch.ppt
http://www.personal.kent.edu/%7Ermuhamma/Compgeometry/MyCG/ConvexHull/GrahamScan/grahamScan.htm
109, 132, 218, 361,
596, 681, 811, 819,
10065, 10078, 10089
10135, 10173,
10256****

Miscellaneous

10095 ****
10209 **
10210

10215 * - Differentiation is enough
10216 **** - Fermat Point
http://en.wikipedia.org/wiki/Fermat_point

10221 *
10228 *** - Greedy Local Search

10242 *

10245 *** - The Closest Pair Problem - Divide and Conquer - O(nlogn) -- You need to learn this Algorithm.
http://www.cs.ucsb.edu/%7Esuri/cs235/ClosestPair.pdf
http://en.wikipedia.org/wiki/Closest_pair_of_points_problem

10251 ***

10478 **

11123
11186 *** looks like geo. but mainly it needs proper use of cumulative arrays.
11529 ***** cumulative + lower_bound

* represents hardness. I don't remember hardness of the problems that are without any star.

Thanks!

--
Nafi