Divide and conquer

Problem:

Given n points in a plane. Every point has a coordinate p(x\_1, y\_1). Want to find the pair of points with minimum distance between them. The distance formula is sqrt(delta\_x^2, delta\_y^2).

Time complexity: O(nlogn)

T(n) = 2T(n/2)+O(n) = O(nlogn)

In each recursive call, check x axis and y axis too see if there are any points with the same x-axis?

Check 11/16 blocks -- too safe

mid point

|  |  |  |  |
| --- | --- | --- | --- |
| c | c | c | c |
| c | c | c | c |
| X4 | c | c | c |
| P1 | X1 | X2 | X3 |

We can check only 7/16 blocks

Sort the 16 points by distance, and check one half with distance over the middle point (delta/2)

|  |  |  |  |
| --- | --- | --- | --- |
| x | x | c | x |
| x | x | c | c |
| x | x | c | c |
| P1 | x | c | c |

Dynamic programming

Goal: give optimal cut of the rod such that the summation of the value is maximized.

Given: a rod of size n, A[1-m] = v1,v2,v3...vm

vi: value of rod with length i

1. recursive formula

in each recursive, the max value of an arbitrary rod of length i.

Known the given first cut, we know the rest is an optimal cut.

1 2 3 4 5 | 6 7 8

first cut

678 is optimal

12345 is optimal

brute force check the first cut (n choices)

...

i++;

max(opt[i]+opt[n-i]);

...