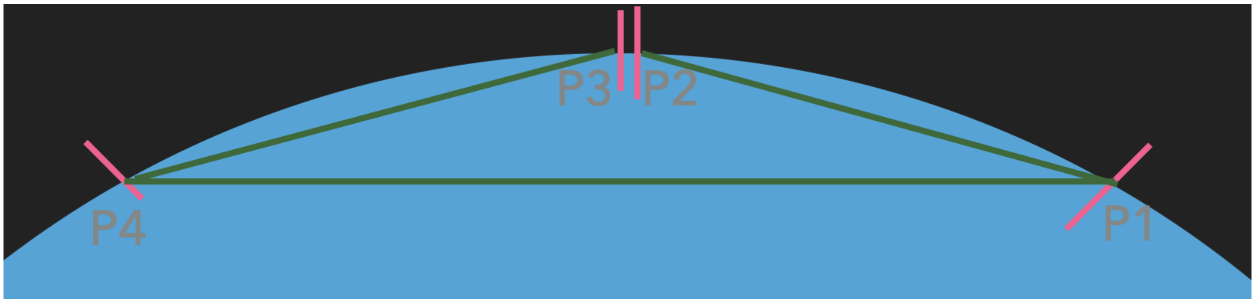
Q1 Many solutions took the median of x_i’s as pivot to divide elements into 2 halves, one half for elements greater than Median and other half all smaller than median, and depending on the sum of probabilities in the smaller half returned the either median of Right half or left half. But Median of the right half or left half of the array is not necessarily the median of entire array, consider the case [(0,0.1),(1,0.1),(2,0.3),(3,0.5)]

Q2:

1. If the correct solution is given with
    1. O(n^4) complexity due to two for loops per element in DP: **10/12**
    2. No mention of crossing chords observation: **9/12**
    3. O(n^2) complexity due to no for loops per element in DP: **8/12**
2. Nearest neighbor solution (Greedy algorithm): **3/12**
   Counter-example:
   This solution does not work in the following case:
   Consider the following arc, where the two points at the top of the arc are very close to each other.
   In such a case, the optimal solution will be to join P1 with P4 and P2 with P3 (as two sides of a triangle will always be more than the third)

   Consider a circle which contains multiple such arcs. In such a case, the greedy solution of neighbors will not work correctly.

3. If the correct idea present but the execution incorrect/incomplete i.e. the DP is not constructed properly or the recursion is not mentioned well: **6/12**
4. Assuming solution for n-2 points and adding 2 new points (with/without 1D DP): **0/12**
   This does not use the crucial property of no crossing chords in the final solution.
5. 2 sets over 1…n and n+1…2n: **0/12**
   This also does not use the crucial property of no crossing chords in the final solution.
6. 2D DP array for n chords vs 2n points: **0/12**
   This algorithm does not work. Either chords of opposing parity don't get considered OR the DP doesn't serve its purpose and algo becomes exponential