

1 point

1. Which of the following is **not** true about KD-trees?
- ☐ It divides the feature space into nested axis-aligned boxes.
  - ☒ It can be used only for approximate nearest neighbor search but not for exact nearest neighbor search.
  - ☐ It prunes parts of the feature space away from consideration by inspecting smallest possible distances that can be achieved.
  - ☐ The query time scales sublinearly with the number of data points and exponentially with the number of dimensions.
  - ☐ It works best in low to medium dimension settings.

1 point

2. Questions 2, 3, 4, and 5 involves training a KD-tree on the following dataset:

	X1	X2
Data point 1	-1.58	-2.01
Data point 2	0.91	3.98
Data point 3	-0.73	4.00
Data point 4	-4.22	1.16
Data point 5	4.19	-2.02
Data point 6	-0.33	2.15

Train a KD-tree by hand as follows:

- First split using X1 and then using X2. Alternate between X1 and X2 in order.
- Use “middle-of-the-range” heuristic for each split. Take the maximum and minimum of the coordinates of the member points.
- Keep subdividing until every leaf node contains two or fewer data points.

What is the split value used for the first split? Enter the exact value, as you are expected to obtain a finite number of decimals. Use American-style decimals (e.g. 0.026).

-0.015

1 point

3. Refer to Question 2 for context.

What is the split value used for the second split? Enter the exact value, as you are expected to obtain a finite number of decimals. Use American-style decimals (e.g. 0.026).

0.995

1 point

4. Refer to Question 2 for context.

Given a query point (-3, 1.5), which of the data points belong to the same leaf node as the query point? Choose all that apply.

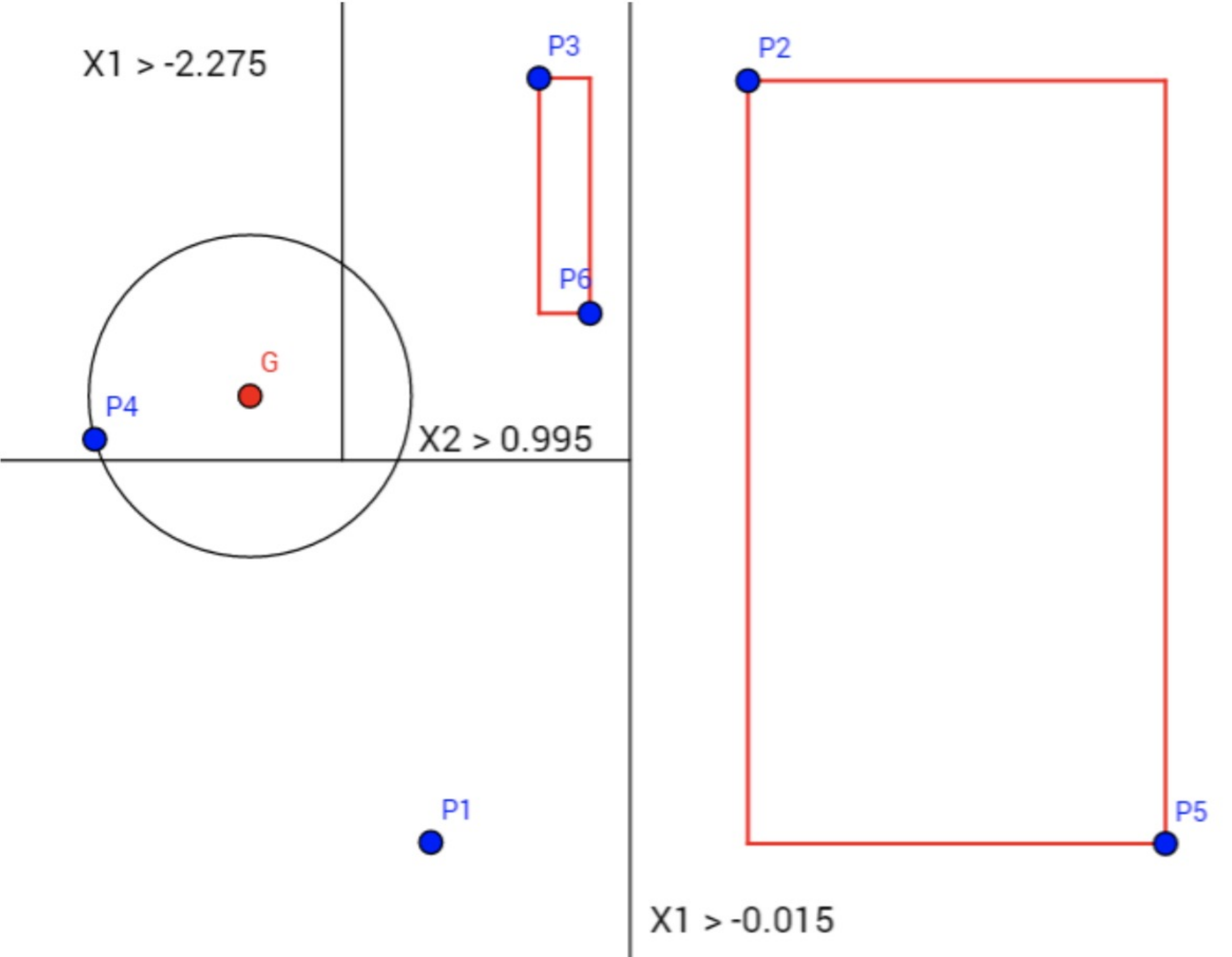
- ☐ Data point 1
- ☐ Data point 2
- ☐ Data point 3
- ☒ Data point 4
- ☐ Data point 5
- ☐ Data point 6

1 point

5. Refer to Question 2 for context.

Perform backtracking with the query point (-3, 1.5) to perform exact nearest neighbor search. Which of the data points would be pruned from the search? Choose all that apply.

Hint: Assume that each node in the KD-tree remembers the **tight bound** on the coordinates of its member points, as follows:



- ☒ Data point 1
- ☒ Data point 2
- ☒ Data point 3
- ☐ Data point 4
- ☒ Data point 5
- ☒ Data point 6