Algorithm 1 ANN search over the augmented neighborhood graph

/* **q**: the query; $\mathscr{X}$: the reference data vectors; $\mathscr{Y}$: the set of bridge vectors; G: the augmented neighborhood graph; Q: the main queue; R: the result set; T: the maximum number of discovered vectors; */

Procedure ANNSearch(**q**, $\mathscr{X}$, $\mathscr{Y}$, G, Q, R, T)

1. /* *Mark each reference vector undiscovered* */
2. **for** each $\mathbf{x} \in \mathscr{X}$ **do**
3. Color[**x**] $\leftarrow$ white;
4. **end for**
5. /* *Extract the nearest bridge vector* */
6. $(\mathbf{y}, D) \leftarrow$ ExtractNextNearestBridgeVector($\mathscr{Y}$);
7. $Q \leftarrow (\mathbf{y}, D)$;
8. $t \leftarrow 0$
9. /* *Start the search* */
10. **while** ($Q \neq \emptyset$ && $t \leqslant T$) **do**
11. /* *Pop out the best candidate vector and expand its neighbors* */
12. $(\mathbf{p}, D) \leftarrow Q.\text{pop}()$;
13. **for** each $\mathbf{x} \in Adj[\mathbf{p}]$ **do**
14. **if** Color[**x**] = white **then**
15. $D \leftarrow \text{dist}(\mathbf{q}, \mathbf{x})$;
16. $Q \leftarrow (\mathbf{x}, D)$;
17. Color[**x**] $\leftarrow$ black; /* *Mark it discovered* */
18. $R \leftarrow (\mathbf{x}, D)$; /* *Update the result set* */
19. $t \leftarrow t + 1$;
20. **end if**
21. **end for**
22. /* *Extract the next nearest bridge vector if* **p** *is a bridge vector* */
23. **if** $\mathbf{p} \in \mathscr{Y}$ **then**
24. $(\mathbf{y}, D) \leftarrow$ ExtractNextNearestBridgeVector($\mathscr{Y}$);
25. $Q \leftarrow (\mathbf{y}, D)$;
26. **end if**
27. **end while**
28. **return** R;

根据优先队列确定前沿数据点，每次扩充到全部邻居。
每个邻居，既加入队列，又更新结果