Algorithm 1 Mondrian Anonymization

```
1: function MondrianAnon(dataset, QIs, k, choose_dimension, single_dimensional)  ▷ n: numero di record, |QIs|: numero di quasi-identificatori
2:     if is_k_anon(dataset, QIs, k) then
3:         return dataset                                                  O(n · |QIs|)
4:     end if
5:     if |QIs| = 0 then
6:         return dataset
7:     end if
8:     if |dataset| ≤ 2k − 1 then
9:         return dataset
10:    end if
11:    dim ← chooseDimension(dataset, QIs, choose_dimension)              O(n · |QIs|)
12:    splitVal ← find_median(dataset, dim)                               O(n log n)
13:    LHS, RHS ← splitDataset(dataset, dim, splitVal)                    O(n) [n], O(n log n) [c]
14:    LHS_copy ← deepcopy(LHS)
15:    RHS_copy ← deepcopy(RHS)
16:    generalize(LHS_copy, dim)                                          O(n)
17:    generalize(RHS_copy, dim)                                          O(n)
18:    for all other_dim ∈ QIs do
19:        if other_dim ≠ dim then
20:            if |LHS_copy| ≤ 2k − 1 then
21:                generalize(LHS_copy, other_dim)                        O(n)
22:            end if
23:            if |RHS_copy| ≤ 2k − 1 then
24:                generalize(RHS_copy, other_dim)                        O(n)
25:            end if
26:        end if
27:    end for
28:    QIsNew ← QIs \ {dim}
29:    if single_dimensional then
30:        return LHS_copy + RHS_copy
31:    end if
32:    left ← MondrianAnon(LHS_copy, QIsNew, k, choose_dimension)
33:    right ← MondrianAnon(RHS_copy, QIsNew, k, choose_dimension)
34:    return left + right
35: end function
```

Operazione	**Complessità**
IS_K_ANON(dataset, QIs, k)	$O(n \cdot \|QIs\|)$
CHOOSEDIMENSION(dataset, QIs, choose_dimension)	$O(n \cdot \|QIs\|)$
FIND_MEDIAN(dataset, dim)	$O(n \log n)$
SPLITDATASET(dataset, dim, splitVal)	$O(n)$ [numerico], $O(n \log n)$ [categorico]
GENERALIZE(dataset, dim)	$O(n)$
Ciclo `ForAll` su $\|QIs\|$	$O(n \cdot \|QIs\|)$

Table 1: Riepilogo delle complessità delle operazioni.