

This algorithm described  $k$ NN query processing by the *Hamming Weight Tree* as is described in paper *Modifying Hamming Spaces for Efficient Search* of authors Vladimir Mic, David Novak, and Pavel Zezula.

---

**Algorithm 1**  $k$ NN query algorithm for the HWT

---

**Input:** query bit string  $q$   
**Output:**  $ret[0, \dots, k-1]$ :  $k$  most similar bit strings to  $q$

```

 $i \leftarrow w(q)$ 
 $ret \leftarrow emptyArray$ 
if  $processNode(L_i, q, 0, ret)$  then ▷ find exact matches
    return  $ret$ 
 $rad \leftarrow 1$  ▷ start incremental search
while  $h(q, ret[k-1]) > rad$  do
    if  $processNode(L_{i-rad}, q, rad, ret)$  then
        return  $ret$ 
    if  $processNode(L_{i+rad}, q, rad, ret)$  then
        return  $ret$ 
     $rad \leftarrow rad + 1$ 
return  $null$  ▷ unreachable state
procedure  $processNode(L_i, q, rad, ret)$ 
    if  $L_i$  is leaf then
        set lower bound of  $L_i$  to  $\infty$ 
        for each  $o \in L_i$  do
            if  $h(q, o) < h(q, ret[k-1])$  then
                update answer  $ret$ 
            if  $h(q, ret[k-1]) = rad$  then
                return  $true$ 
        return  $false$ 
    Sort children  $L_c$  of  $L_i$  according to lower bound
    for each  $L_c$  with lower bound  $\leq rad$  do
        if  $processNode(L_c, q, rad, ret)$  then
            return  $true$ 
    set lower bound of  $L_i$  to the minimal lower bound provided by its children
    return  $false$ 

```

---