```
algo(T, h1, h2, n1, n2, k1, k2, F, h)
      n = 0
      current = 0

      if (T != NIL)
            nsx = ALGO(sx[T], h1, h2, n1, n2, k1, k2, T, h+1)
            ndx = ALGO(dx[T], h1, h2, n1, n2, k1, k2, T, h+1)
            n = nsx + ndx
            if ((key[T] >= k1) && (key[T] <= k2)) //compreso tra k1 e k2
                  current = 1
                  if ((key[T] % 2) = 0)   //è pari
                        if ((h >= h1) && (h <= h2))
                              if ((n >= n1) && (n <= n2))
                                    cancella(T, F);
      return n + current

----------

staccaMin(T, F)
      if (T != NIL)
            if (sx[T] != NIL)
                  return staccaMin(sx[T], T)
            else //trovato
                  if (T = sx[F])
                        sx[F] = dx[T]
                  else
                        dx[F] = dx[T]
      return T

----------

cancella(T, F)
      nodo = T
      if (sx[T] = NIL)
            if (T = sx[F])
                  sx[F] = dx[T]
            else
                  dx[F] = dx[T]
      else if (dx[T] = NIL)
            if (T = sx[F])
                  sx[F] = sx[T]
            else
                  dx[F] = sx[T]
      else
            nodo = staccaMin(dx[T], T)
            key[T] = key[nodo]
      dealloca[nodo]
```