Algorithm 1 DESIGN - search over the number of cuts K

Require: alignment A, weights w, library cap C_L, nucleotide cap C_N
Ensure: a design maximizing covered weight, or NONE

procedure DESIGN(A, w, C_L, C_N)
  $F \leftarrow \emptyset$ ▷ *frontier: best solution at each K*
  for $K = 1, \ldots, K_{\max}$ **do**
    $\mathcal{C} \leftarrow$ PLACECUTS(A, K) ▷ *candidate cut sets*
    $s^\star_K \leftarrow$ NONE
    for $(\kappa, \tau) \in \mathcal{C}$ **do**
      ▷ *greedily build a solution for this fixed set of cuts*
      $s \leftarrow$ GREEDY(A, w, κ, τ, C_L, C_N)
      if $s =$ NONE **or** s spans a forbidden site (BsmBI) at a junction **then**
        skip
      end if
      $W(s) \leftarrow \sum_{c \in \text{covered}(s)} w(c)$ ▷ *weight of the encoded natural sequences*
      $\text{nt}(s) \leftarrow$ total length in nucleotides, incl. Golden Gate overhead
      ▷ *prioritise coverage; break ties on fewest nucleotides*
      if $\big(W(s),\, -\text{nt}(s)\big) \succ_{\text{lex}} \big(W(s^\star_K),\, -\text{nt}(s^\star_K)\big)$ **then**
        $s^\star_K \leftarrow s$
      end if
    end for
    if $s^\star_K \neq$ NONE **then**
      $F[K] \leftarrow s^\star_K$ ▷ *best solution at this K*
    end if
  end for
  if $F = \emptyset$ **then**
    return NONE
  end if
  return $\arg\max_{s \in F} \big(W(s),\, -\text{nt}(s),\, -K(s)\big)$ ▷ *lexicographic*
end procedure

Algorithm 2 GREEDY - extend a solution for a fixed set of cuts

Require: alignment A, weights w, cuts κ, junction tokens τ, library cap C_L, nucleotide cap C_N
Ensure: layers, coverage, weight, stop reason

procedure GREEDY(A, w, κ, τ, C_L, C_N)
    ▷ *a layer holds the fragments at one fragment position;*
    ▷ *for K cuts there are K layers, and layer f lists its fragments*
    $L \leftarrow$ one unit per layer, seeded from the heaviest core
    (cov, W, lib) $\leftarrow$ COMPUTESTATS(L)
    while an addition is possible **do**
        $m^\star \leftarrow$ NONE
        for each core $c \notin$ cov **do**
            ▷ *cheapest modification of each layer that would encode c*
            $p \leftarrow \left[\text{cheapest move in layer } f \text{ for } c \;\; \forall f\right]$ ▷ *reuse | degen codon | add fragment*
            (lib$'$, cov$'$, nt$'$) $\leftarrow$ PRICE(p)
            $g \leftarrow W(\text{cov}') - W$ ▷ *gain in covered weight*
            if $g \leq 0$ **then**
                skip
            end if
            if lib$' > C_L$ **or** nt$' > C_N$ **then**
                record blocking cap; **skip**
            end if
            $\sigma \leftarrow (\text{lib}' - \text{lib})/g$ ▷ *library cost per unit of weight gained*
            if $(\sigma, \text{nt}', -g, c) \succ_{\text{lex}} m^\star$ **then** ▷ *equivalent to maximizing gain / library cost*
                $m^\star \leftarrow p$ ▷ *implemented for numerical stability*
            end if
        end for
        if $m^\star =$ NONE **then**
            record stop reason; **break**
        end if
        ▷ *apply the move*
        $L \leftarrow L \cup \{\text{new fragments of } m^\star\}$
        (cov, W, lib) $\leftarrow$ COMPUTESTATS(L)
    end while
    return (L, cov, W, stop reason)
end procedure