# 2020牛客暑期多校训练营（第一场）

郭晓絮
shyhaimao@twitter

# B–Suffix Array

- Let C_i = min_{j > i and s_j = s_i} {j – i}
- The B–Suffix Array is equivalent to the suffix array of C_1 C_2 … C_n

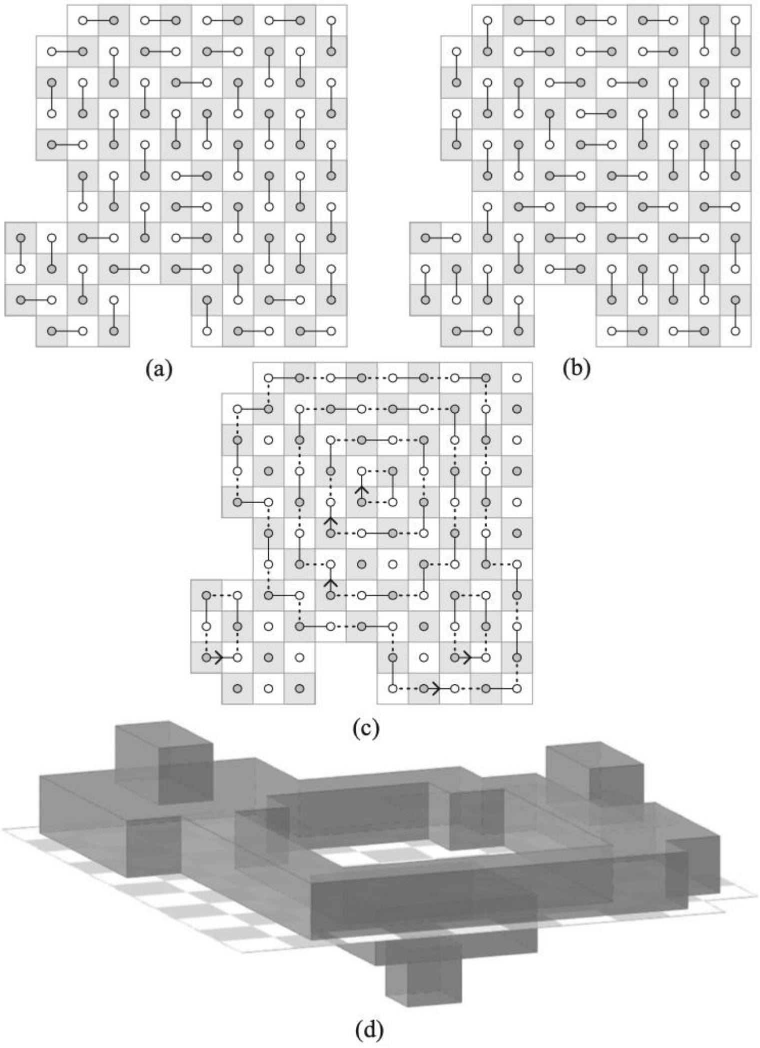
- Detailed proof can be found in “Parameterized Suffix Arrays for Binary Strings
- ” http://www.stringology.org/event/2008/p08.html

# Infinite Tree

- First, compute the "virual tree" of {1!, 2!, ..., n!}
- Second, to compute the actual cost, use Segment Tree or Fenwick Tree.
- O(m log^2 m)

# Domino

- See “Distances in Domino Flip Graphs”



# Quadratic Form

- The answer is b^T A^{-1} b, which can be proved by Lagrange Duality.
- All we need is to compute the inverse matrix of the matrix A.

# Counting Spanning Trees

- The number of spanning trees is prod_{i >= 2} deg(x_i) deg(y_i)
- Detailed proof can be found in “Enumerative properties of Ferrers graphs” https://arxiv.org/pdf/0706.2918.pdf

# Infinite String Comparision

- Compare the string a^{infty} and b^{infty} directly
- By the Periodicity Lemma, if there is no mismatches in the first a + b – gcd(a, b) characters, the two string are identical

# BaXianGuoHai, GeXianShenTong

- **For simplicity, we denote the multiplication as +, and exponentiation as $*$**
- **Precompute $B_{i, j} = 2^{W * j} * v_i$**
- **To compute $\sum_{i, j} (\sum e'_{i, j} * 2^{W * j})\, v_i$**
  - $= \sum_x x \sum_{i, j} [e'_{i, j} = x]\, B_{i, j} = \sum_x x\, Q_x$
- **To compute $\sum_x x\, Q_x$**
  - $= \sum_x (\sum_{y \geq x} Q_y)$
- **The overall complexity is $O(nm / W + 2^W)$**
- **Taking $W = 16$ yields a fast enough solution**

# Minimum-cost Flow

- We denote the cost in a network with capacity c and flow f as cost(c, f).
- cost(c, 1) = cost(c * 1/c, 1 / c) * c = cost(1, 1 / c) * c
- For a network with unitary capacity, its cost grows linearly with the flow f, with at most O(m) pieces.
- Thus, we can compute O(m) pieces first, and query in O(log m) time.

# 1 or 2

- **For an edge $e=(x, y)$ where $d\_x = d\_y = 2$, add the following edges:**
  - $(x, e)$ $(x', e)$
  - $(y, e')$ $(y', e')$
  - $(e, e')$
- **The problems is turned into to find a perfect matching in a general graph, which can be solved with Edmond's Algorithm.**

# Easy Integration

- The value is (n!)^2 / (2n+1)!
- Detailed proof can be found in “Wallis' integrals”.
  https://en.wikipedia.org/wiki/Wallis%27_integrals

牛客竞赛
AC.NOWCODER.COM

# Thanks

牛客竞赛
AC.NOWCODER.COM