Finding the Largest Fixed-Content Necklace

Matthew Ng
Supervisor: Joe Sawada
CIS*4910

University of Guelph

April 2022

Intro

Our goal: A polynomial time algorithm to find the largest fixed-content necklace.

- Key step in ranking and unranking fixed-content Lyndon words and necklaces
- Applications in cut-down de Bruijn sequence generation
- Generalize a previous algorithm in fixed-density necklace generation

Result: A $O(n^2)$ algorithm that generates the larged fixed-content necklace under word-RAM model.
Key definitions:

- A **necklace** is the lexicographically smallest element in a set of all rotations of a string.
- **Fixed-content** refers to a necklace with a specified number of occurrence of symbols.

Summary of improvements from last time

- Sharpened notation in paper to aid in understanding/solving problem
- Developed a new optimization for algorithm in the case of majority 0 content
- Found the realm of the general multiway-partition problem as a very helpful aid
- Simplified aspects of algorithm, making implementation easier
- Starting to formalize lemmas, form proofs

Examples of fixed-content necklaces

Example

Given a content $[0, 0, 1, 2, 2, 2]$:
021022, 020221, 020212, 012202, 012022, 010222, 002221, 002212, 002122, 001222

- Here is a list of every possible necklace given such fixed-content. In this list, the largest necklace is denoted in bold.
- Our algorithm seeks to return this necklace given the content.

General approach

We devise an recursive algorithm defined as $LargestNeck(C)$ that

- Recursively multi-way partitions content into simplified content until it hits a base case such that the ordering is trivial
- We decode the previously simplified content as the properly ordered necklace is moved up the recursive stack

Multi-way partitioning

Multi-way partitioning is a general problem that seeks to distribute a content of numbers through a specified number of partitions and optimize the partitions making up three objective functions:

1. Minimize the largest of the k subset sums.
2. **Maximize the smallest of the k subset sums.**
3. Minimize the difference between the largest and smallest of the k subset sums.

ie; $S = \{13, 9, 9, 6, 6, 6\}$ and $k = 3$, $\{13, 6\}, \{9, 6\}, \{9, 6\}$, smallest subset sum is 15, which is the maximal of this content.

- In our case, we use a slightly modified version of objective function 2, where we maximize lexicographically instead of sum wise.

Definitions

- Let C be the content of a necklace, denoted by a list of symbols
- Let z be the number of 0s in C
- Let n be the length of a necklace

Our way of multi-way partitioning

- Goal: Lexicographically maximize the lexicographically smallest string out of z distributed strings.
- Returns: A list of tuples, S_z, first index is the associated lexicographical rank and second index is the associated new distributed string.

We denote this function as $\mathit{MultiwayPartition}(C, z)$.

- A new content is C' is then made up of the rankings and a map f is use to map the new content to it's constituent string.
- Since only one constant time operation is needed for each symbol, it runs in $O(n)$ time.

Example of multi-way partitioning

Example

C = [5,5,5,5,5,5,5,4,4,4,4,4,4,4,4,3,3,3,2,2,2,1,1,1,1,0,0,0,0,0,0].
$z = 6$ lists are created, serving as the partitions of content.

$$[5, 5], [5], [5], [5], [5], [5]$$

All 5s are evenly distributed. Start from the second list.

$$[5, 5], [5, 4, 4], [5, 4, 4], [5, 4, 4], [5, 4], [5, 4]$$

$$[5, 5], [5, 4, 4], [5, 4, 4], [5, 4, 4], [5, 4, 3, 3], [5, 4, 3]$$

$$[5, 5], [5, 4, 4], [5, 4, 4], [5, 4, 4], [5, 4, 3, 3], [5, 4, 3, 2, 2, 2, 1, 1, 1, 1]$$

All remaining non-zero content is then distributed into the last list.
$S_z = [(0, 5432221111), (1, 5433), (2, 544), (2, 544), (2, 544), (3, 55)]$
$C' = [3, 2, 2, 2, 1, 0], f = \{0 : 5432221111, 1 : 5433, 2 : 544, 3 : 55\}$

Overall recurrence for the LargestNeck(C)

$$
LargestNeck(C) = \begin{cases} 0w_1 0w_2 \cdots 0w_z & \textit{if } 0 < z \leq \frac{n}{2} \\ 0s_1 0s_2 \cdots 0s_j & \textit{if } 0 < \frac{n}{2} < z \\ 0 \cdot C[n-1] \cdot C[n-2] \cdots C[1] & \textit{if } z = 1 \\ 0^n & \textit{if } k = 1 \end{cases}
$$

where $\{w_1, w_2, \ldots, w_z\} = MultiwayPartition(C, z)$ and $\{s_1, s_2, \ldots, s_j\} = MultiwayPartition(C, z)$.

Example of finding fixed-content necklace

Examples

Let $C = [2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0]$.
$0 < z \leq \frac{n}{2}$, first case of $MultiwayPartition(C, z)$ forms lists:

$$[2, 2], [2, 2], [2, 1, 1, 1], [2, 1, 1, 1]$$

$S_z = [(0, 2111), (0, 2111), (1, 22), (1, 22)], f = \{0 : 2111, 1 : 22\}, C' = \{1, 1, 0, 0\}$. $LargestNeck(C')$, Let R be new content, x be number of zeroes. First case of $MultiwayPartition(R, x)$ arises again.

$$[1], [1]$$

$T_z = [(0, 1), (0, 1)], g = \{0 : 1\}, R' = \{0, 0\}$. Next recursive call, basecase of $k = 0$ is hit. 00 is returned up the recursive stack which is decoded to

$$0101$$

using g. 0101 then decodes to

$$0211102202111022$$

Final remarks

Since each recursive call eliminates at least one symbol from the old content, order of n many $O(n)$ recursive calls are many, giving this algorithm a running time of $O(n^2)$.

- Future Work
  - Extend to Lyndon words
  - Work on proofs in the paper
  - Solve for problem of finding largest fixed-content necklace less than or equal to necklace of same length
  - Extend this paper to rank and unrank fixed-content necklaces and Lyndon words
  - Possible application of multi-way partitioning solving for some $k - ary$ cases of cut-down de Bruijn sequence construction

Thanks for attending!