

Suffix Tree

จุดทำโนด

Group ถ้าหากก ก , Group กุนหน่าวา





ក៉ាវប៉ូនាំសេនօ

- Suffix Tree គឺអែនី?
- Data structure នៃការកែប្រើប្រាស់បានបង្ហាញបានល្អ
- Algorithm ជីថែរមានី?
- ប៉ាស៊ុនត៉ុវត្ថុការទាំងនេះ មិនតាមគឺជាប្រព័ន្ធឌឹងទៀត

Suffix Tree คืออะไร?

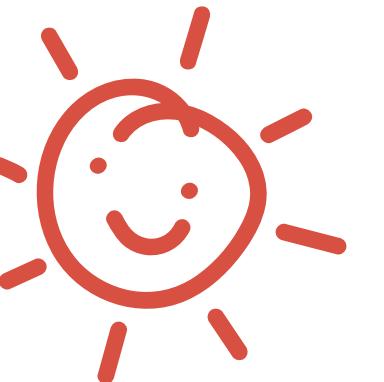


Suffix Tree

เป็นโครงสร้างข้อมูลแบบต้นไผ้ที่ใช้เก็บชุดของ suffixes (ส่วนท้ายของลตรง) ของลตรงที่กำหนด ซึ่งช่วยให้สามารถค้นหาลตรงได้อย่างมีประสิทธิภาพ

- Longest Repeated Substring (LRS)
- Longest Common Substring (LCS)

การเก็บข้อมูลของ Suffix Tree



Suffix Tree เป็นโครงสร้างข้อมูลที่ดีที่สุดสำหรับการค้นหา LRS และ LCS
เนื่องจากสามารถค้นหาล่วงที่ซ้ำกันได้ใน $O(n)$

ข้อดี:

- การค้นหาทำได้เร็วในเวลา $O(n)$
- ใช้ได้ดีกับสตริงขนาดใหญ่

ข้อเสีย:

- ต้องใช้พื้นที่ $O(n)$ ซึ่งค่อนข้างมาก



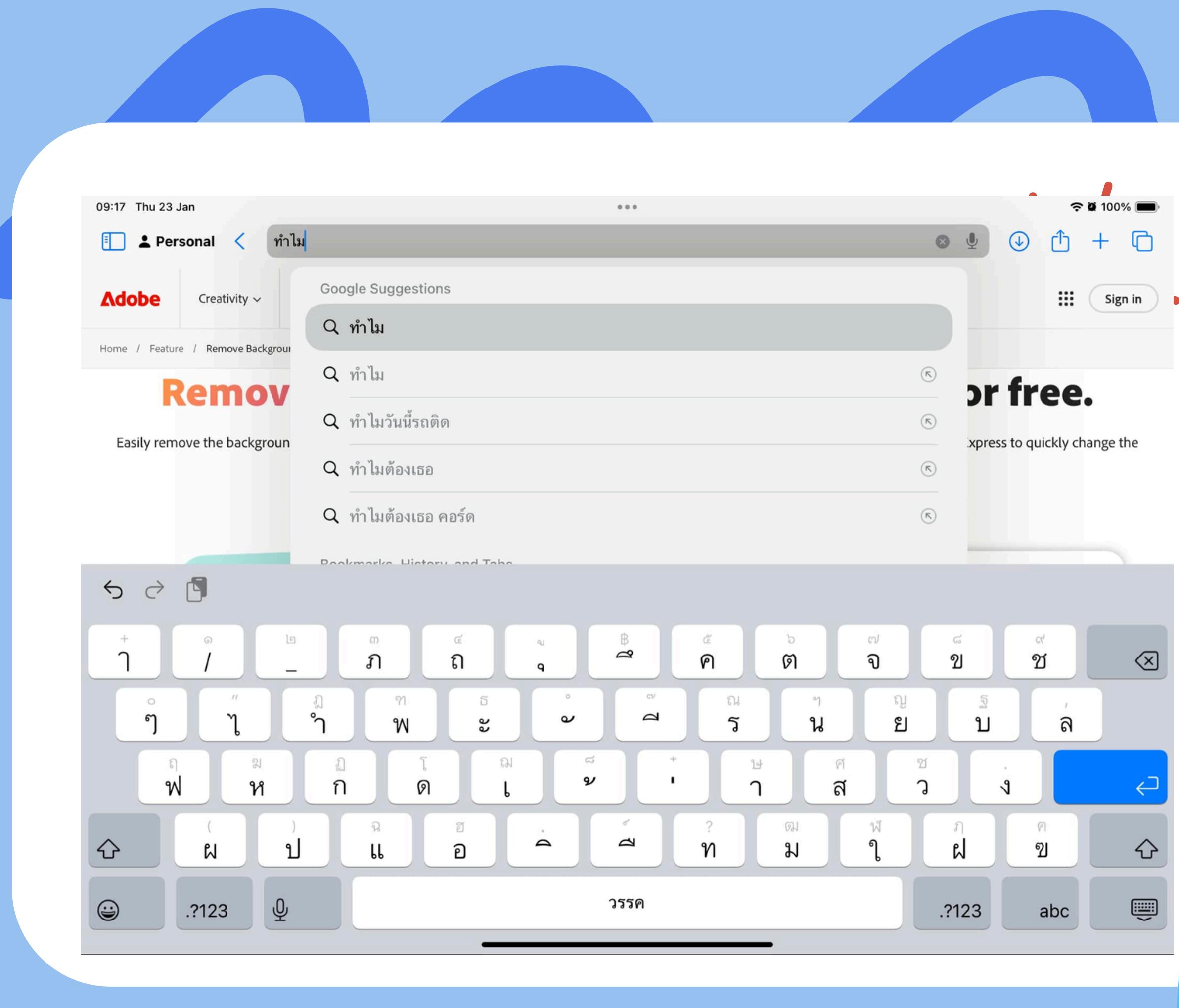
Suffix Tree ຜົນເປົາກາໂວໄສ

ອັລກອຣີຮົມໜີຜົນເປົາກາເຊ ຄືວ່າ *substring* ທີ່ຂ້າກັນ ດາວທີ່ສຸດ ໃນລຕິງ n ທີ່ກຳນົດ

ກລັກກາຣທຳງານ

- ໃຫ້ Suffix Tree ໃນກາຣຊ່ວຍຄັນຫາ
- ຂາຕໍາແນ່ງ *substring* ທີ່ຂ້າກັນຈາກ Internal Nodes ທີ່ມີລູກມາກວ່າ 1
- ເລືອກ Internal Node ທີ່ອຸ່ນລຶກທີ່ສຸດເປັນຄຳຕອບ

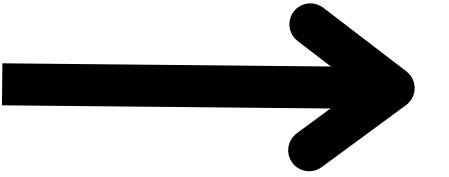




Suffixes ของ "banana\$" (ເຕັນ \$ ເພື່ອນິ້ນບອກຈຸດສິນສຸດ)

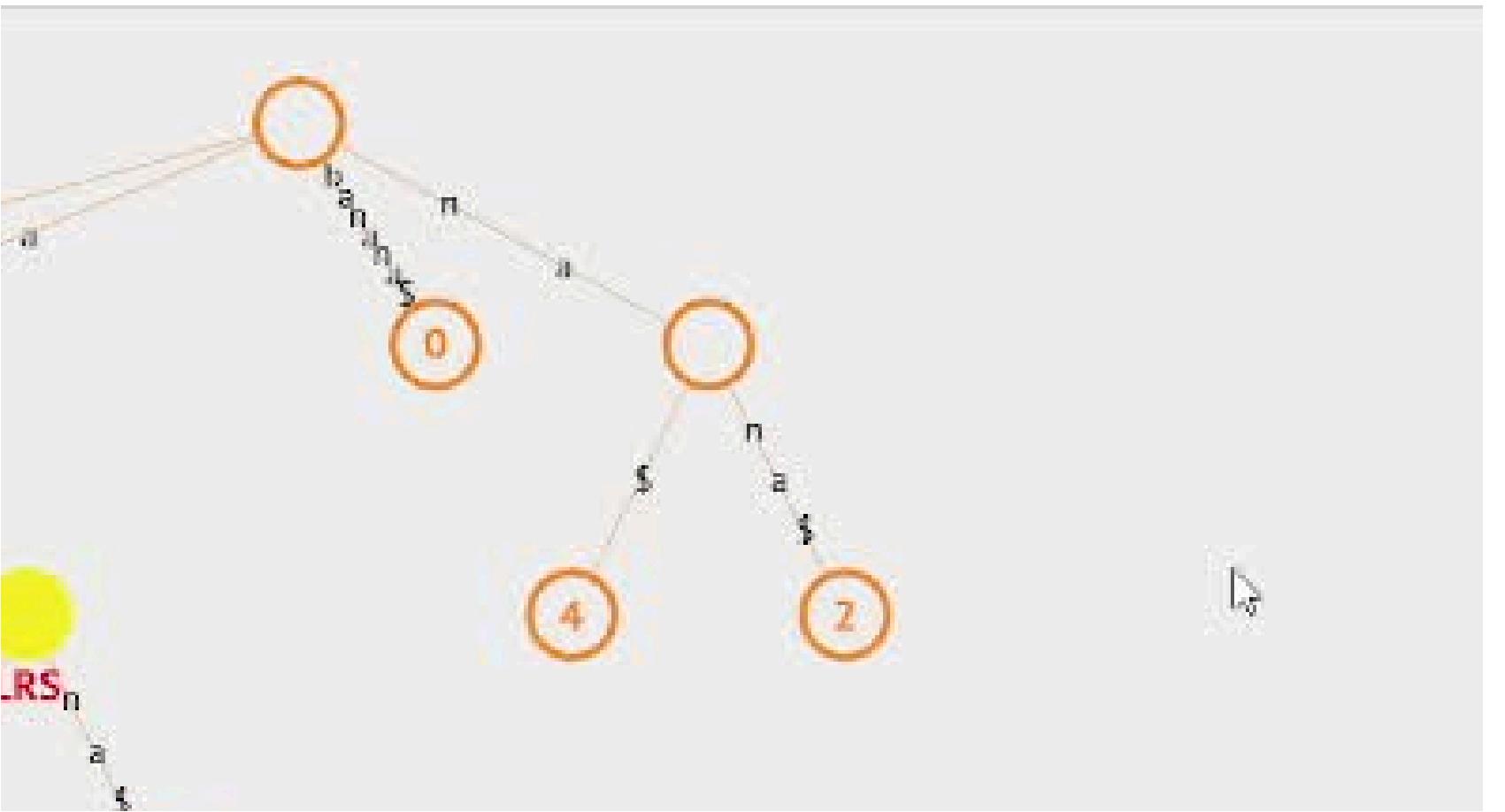
Suffixes:

1. banana\$
2. anana\$
3. nana\$
4. ana\$
5. na\$
6. a\$



Suffixes:

1. a\$
2. ana\$
3. anana\$
4. banana\$
5. na\$
6. nana\$



ນໍາເລັນວາ
ຕົວອຕ່າງການ
ທຳງານ



Suffixes ของ "banana \$" , "nana#" (เติม \$, # เพื่อบ่งบอกจุดสิ้นสุด)

Suffixes:

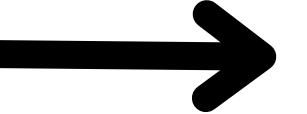
1. banana\$
2. anana\$
3. nana\$
4. ana\$
5. na\$
6. a\$



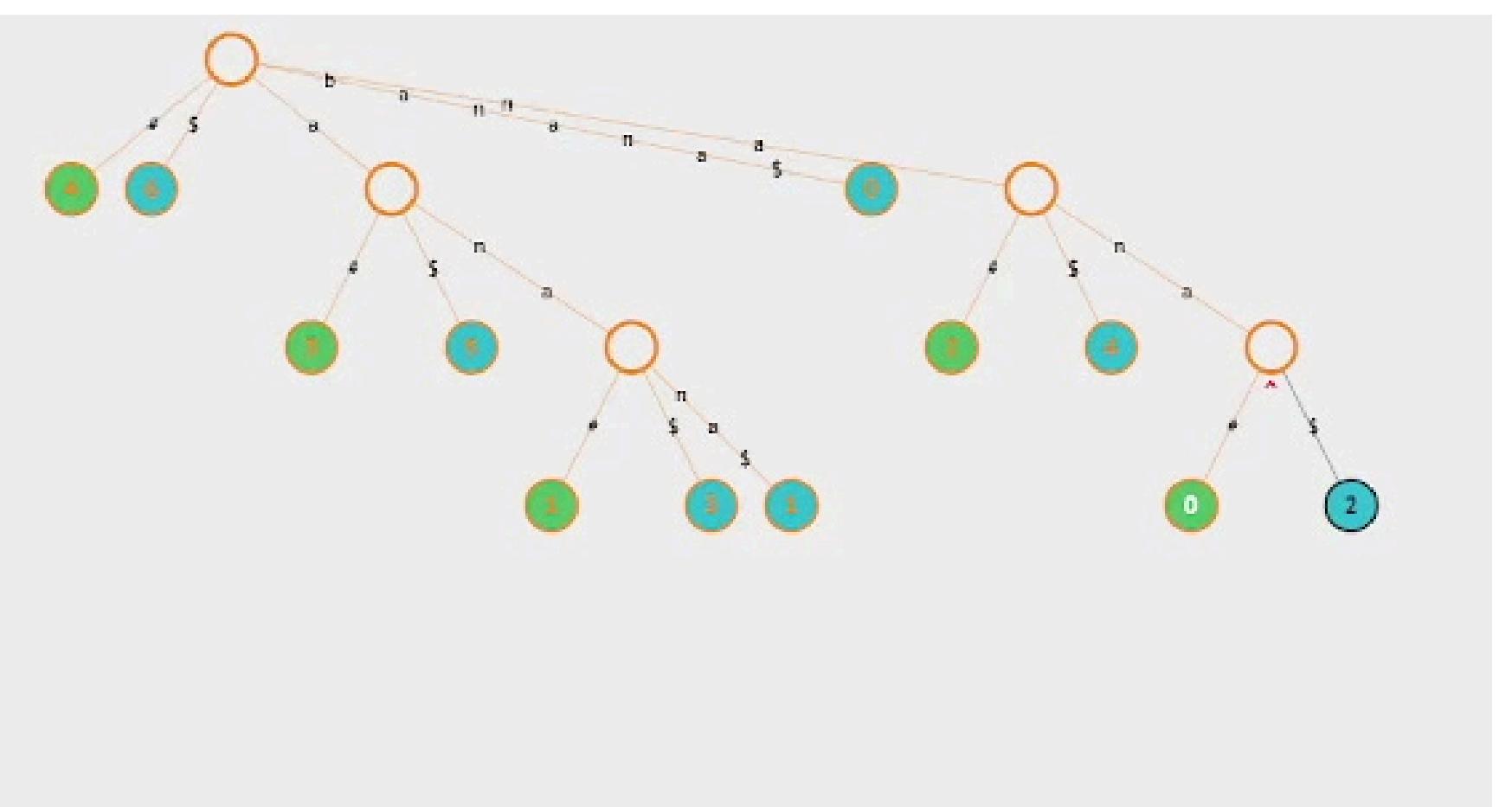
Suffixes:

1. a\$
2. ana\$
3. anana\$
4. banana\$
5. na\$
6. nana\$

- Suffixes:
1. nana#
 2. ana#
 3. na#
 4. a#



- Suffixes:
1. a#
 2. ana#
 3. na#
 4. nana#



นำเสนองาน ตัวอย่างการ ทำงาน



สรุป

- **Longest Repeated Substring (LRS)** → ใช้ค้นหาส่วนที่ซ้ำในลตรง
- **Longest Common Substring (LCS)** → ใช้กับสองลตรงที่รวมกันในต้นไปเดียว



ចុហារ បំរសណ៍

