**Hashing**

**Palíndromos:**

* Calculando el hash del string normal y del inverso

**Borrar pedazos Strings:**

* Con las sumas parciales calculo las dos partes resultantes del String

**Comparar substrings de una palabra (versión hardcore):**

* Tengo un string donde tiene signos de interrogación y significan que yo puedo reemplazar ese signo de interrogación por cualquier letra. Estudiar MATCHING: HOPCROFT-KARP

**Hashear caminos de árboles:**

* Un árbol donde los nodos se guardan los hash de las palabras construidas. Las aristas son letras

**Aho Corasick**

**TRIE:** Representación de un conjunto de palabras con la menor cantidad de nodos

**Encontrar todas las ocurrencias de un conjunto de strings en un texto**

* Tengo que hacer la función go por cada nodo que visito y hacer dp donde se guarda el valor de apariciones por cada nodo y su sufix link

**Encontrar el mínimo string que contenga un conjunto de string como substrings**

* Utilizamos máscaras ds(vi, mask) -> dp. Usamos un bfs donde queue((vi, mask))

**Suffix automata**