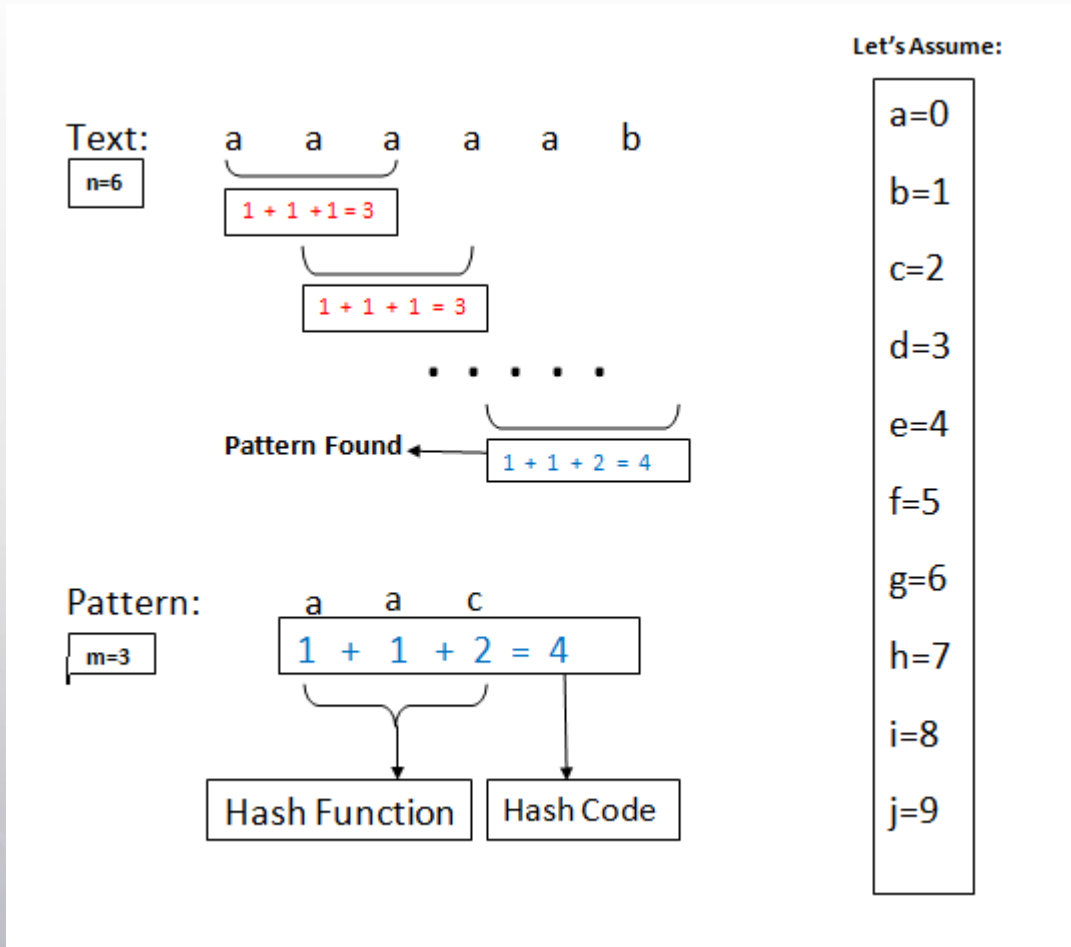




# 라빈 카프 알고리즘이란?



해시 기법을 통한 문자열 알고리즘

✓ 간단한 구조를 통한 구현으로 문자열의 비교가 가능함

✓ 다른 비교 알고리즘과 비교하여 빠른 성능을 보인다.

해싱: 많은 양의 값들을 해시 함수  $H(x)$ 에 대입하여 나온 결과를 인덱스로 하여 해시테이블에 저장하는 방식으로 값을 보존하는 자료구조.  
긴 데이터를 그것을 상징하는 짧은 데이터로 바꿔주는 것

# 라빈 카프에서 해시 기법이란?

## 예시

문자열 'abbac'에 대하여 해시는

아스키 코드 값  $a \rightarrow 97$  /  $b \rightarrow 98$  /  $c \rightarrow 99$  에 대하여

abbac 해시 값

$$\begin{aligned} &= 97 * 2^4 + \\ &\quad 98 * 2^3 + \\ &\quad 98 * 2^2 + \\ &\quad 97 * 2^1 + \\ &\quad 99 * 2^0 \\ &= 4,573 \end{aligned}$$

각 문자의 아스키코드 값에 2의 제곱수를 차례대로 곱하여 모두 더해준 값.  
일반적으로 다른 문자의 경우 다른 해시값이 나옴.

# 라빈 카프에서 해시 기법이란?

## 예시

문자열 'abbac'에 대하여 해시는

아스키 코드 값  $a \rightarrow 97$  /  $b \rightarrow 98$  /  $c \rightarrow 99$  에 대하여

abbac 해시 값

$$\begin{aligned} &= 97 * 2^4 + \\ &\quad 98 * 2^3 + \\ &\quad 98 * 2^2 + \\ &\quad 97 * 2^1 + \\ &\quad 99 * 2^0 \end{aligned}$$

$$= 4,573$$

간단하게 각 문자의 아스키 코드 값에 2의 제곱수를 차례 대로 곱하여 더해주어 해시 값을 만들어 냈.

이를 통해 서로 다른 문자열에 대해 다른 해시 값을 가진 문자열 생성 가능.

# 라빈 카프에서 해시 기법이란?

예시

문자열 'abbac'에 대하여 해시는

abbac 해시 값

$$\begin{aligned} &= 97 * 2^4 + \\ &\quad 98 * 2^3 + \\ &\quad 98 * 2^2 + \\ &\quad 97 * 2^1 + \\ &\quad 99 * 2^0 \end{aligned}$$

$$= 4,573$$

abbcc 해시 값

$$\begin{aligned} &= 97 * 2^4 + \\ &\quad 98 * 2^3 + \\ &\quad 98 * 2^2 + \\ &\quad 99 * 2^1 + \\ &\quad 99 * 2^0 \end{aligned}$$

$$= 4,577$$

간단하게 각 문자의 아스키 코드 값에 2의 제곱수를 차례 대로 곱하여 더해주어 해시 값을 만들어 냈.

이를 통해 서로 다른 문자열에 대해 다른 해시 값을 가진 문자열 생성 가능.

# 라빈 카프에서 해시 기법이란?

예시

문자열 'abbac'에 대하여 해시는

abbac 해시 값

$$\begin{aligned} &= 97 * 2^4 + \\ &98 * 2^3 + \\ &98 * 2^2 + \\ &97 * 2^1 + \\ &99 * 2^0 \end{aligned}$$

$$= 4,573$$

abbcc 해시 값

$$\begin{aligned} &= 97 * 2^4 + \\ &98 * 2^3 + \\ &98 * 2^2 + \\ &99 * 2^1 + \\ &99 * 2^0 \end{aligned}$$

$$= 4,577$$

비교하고자 하는 문자열과 부분 문자열의 해시 값이 일치 할 때만 문자열을 재검사 하여  
일치여부 확인

[효율성 증가]

# 문자열 비교 예시

예시

긴글 : ababacabacaabacaaba

부분 문자열 : abacaaba

a	b	a	b	a	c	a	b	a	c	a	a	b	a	c	a	a	b	a
a	b	a	c	a	a	b	a											

긴글 해시 : 24824  
부분 해시 : 24833

불일치 [넘어감]

# 문자열 비교 예시

## 예시

긴글 : ababacabacaabacaaba

부분 문자열 : abacaaba

a	b	a	b	a	c	a	b	a	c	a	a	b	a	c	a	a	b	a
a	b	a	c	a	a	b	a											

긴글 해시 : 24913  
부분 해시 : 24833

불일치 [넘어감]



# 문자열 비교 예시

## 예시

긴글 : ababacabacaabacaaba

부분 문자열 : abacaaba

a	b	a	b	a	c	a	b	a	c	a	a	b	a	c	a	a	b	a
		a	b	a	c	a	a	b	a									

긴글 해시 : 24937  
부분 해시 : 24833 불일치 [넘어감]

비교하고자 하는 문자열과 부분 문자열의 해시 값이 일치 할 때만 문자열을 재검사 하여 일치여부 확인

반복 ...

# 문자열 비교 예시

## 예시

긴글 : ababacabacaabacaaba

부분 문자열 : abacaaba

a	b	a	b	a	c	a	b	a	c	a	a	b	a	c	a	a	b	a
						a	b	a	c	a	a	b	a					

긴글 해시 : 24833  
부분 해시 : 24833 일치

해시 값이 일치 함으로 차례대로 부분 문자열과 매칭하여 검사

# 라빈 카프 강점

구현이 간단함 (해시화 및 값 비교를 통한 간단한 구조)

예외 상황 (해시 값이 중복되는 Collision, 해시 값 일치 비교를 위해 필요한 MOD연산 필요성)의 발생빈도가 크지 않고 간단히 해결 가능함

일반적인 성능이 보장 됨  $O(n)$

긴 글 해시를 구하는 과정이 수학적 반복으로 쉽게 풀이 됨  
[왜 인지 관련 내용 고찰에 작성해 볼 것]



## Ref.

- Cormen, Thomas H.r, et al.,auths. Introduction to Algorithms Cambridge: MIT Press, 1997
- <https://blog.naver.com/ndb796/221240679247>
- Go2Net Website for String Matching Algorithms
  - [www.go2net.com/internet/deep/1997/05/14/body.html](http://www.go2net.com/internet/deep/1997/05/14/body.html)
- Yummy Yummy Animations Site for an animation of the Rabin-Karp algorithm at work
  - [www.mills.edu/ACAD\\_INFO/MCS/CS/S00MCS125/String.Matching.Algorithms/animations.html](http://www.mills.edu/ACAD_INFO/MCS/CS/S00MCS125/String.Matching.Algorithms/animations.html)
- National Institute of Standards and Technology Dictionary of Algorithms, Data Structures, and Problems
  - [hissa.nist.gov/dads/HTML/rabinKarpAlgo.html](http://hissa.nist.gov/dads/HTML/rabinKarpAlgo.html)