

TS 문자열 게임

김 태 현

21.09.25_문자열관리프로그램

문제 개요

긴 메인 문자열 존재, 아래 네가지 명령 수행

1. push_back : 메인 문자열 뒤에 문자열 추가
2. pop_back : 메인 문자열의 뒤에서 몇개의 문자 제거
3. reverseStr : 메인 문자열을 뒤집음
4. getCount : 메인 문자열 내에 특정 문자열이 몇개 존재하는지 파악

[제약사항]

1,3,4 함수 호출 각 30,000회

2 함수 호출 100회

초기 메인문자열 길이 $\leq 30,000$

추가, 삭제, 검색 문자열 길이 ≤ 4

문제 분석

- 메인 문자열의 최대 길이 : $30,000 + 4 * 30,000 = 150,000$
- 가장 naïve한 접근시,
 - 1) **pushBack()**
문자열 맨 뒤에 최대 4자리 문자열 추가, $O(4) * 30,000$ 회
 - 2) **popBack()**
문자열 맨 뒤에서 최대 len개의 문자열 삭제, $O(150,000)$
 - 3) **reverseStr()**
문자열 뒤집기, $O(150,000) * 30,000$ 회
 - 4) **getCount()**
4자리 문자열 일치하는 개수 검색, $O(150,000 * 4) * 30,000$ 회

=> reverseStr(), getCount() 함수 최적화 필요

해법 연구

reverseStr() 처리

- 실제로 뒤집는게 불가능
- `bool rev` 를 활용하여 뒤집어진 상태인지 표시
- 뒤집어졌다면 `push`, `pop`을 앞쪽에서 진행

rev = 0, 올바른 상태

a b c d e

“abcde”



reverseStr()

rev = 1, 뒤집힌 상태

a b c d e

“edcba”



pushBack(“fgh”)

rev = 1, 뒤집힌 상태

h g f a b c d e

“edcbafgh”

해법 연구

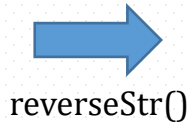
reverseStr() 처리

- 실제로 뒤집는게 불가능
- `bool rev` 를 활용하여 뒤집어진 상태인지 표시
- 뒤집어졌다면 `push`, `pop`을 앞쪽에서 진행
- `getCount()` 시에 `rev=1` 이면, `mainStr`을 뒤집는게 아니라 `subStr`을 뒤집어서 처리

rev = 0, 올바른 상태

a b c d e

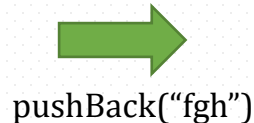
"abcde"



rev = 1, 뒤집힌 상태

a b c d e

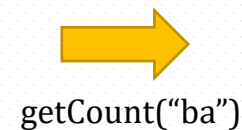
"edcba"



rev = 1, 뒤집힌 상태

h g f a b c d e

"edcbafgh"



rev = 1, 뒤집힌 상태

h g f a b c d e

"edcbafgh"

a b

"ba"

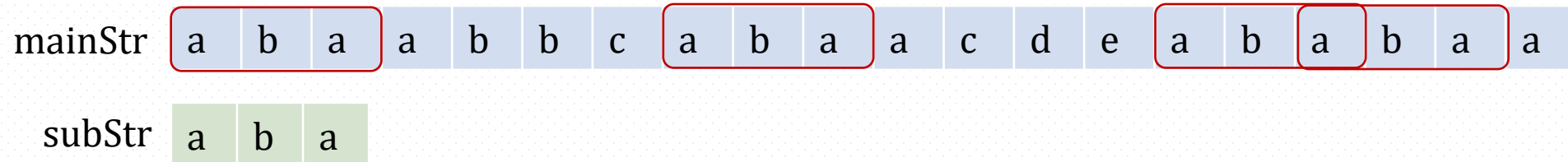
해법 연구

getCount(char subStr[]) 처리

: 메인 문자열의 모든 substring을 검색 하면 시간초과 => 검색 최적화 필요

검색 최적화 keyword

- subStr과 “**일치**”하는 substring 개수 파악
- subStr의 길이 “**최대 4**”



해법 연구

getCount(char subStr[]) 처리

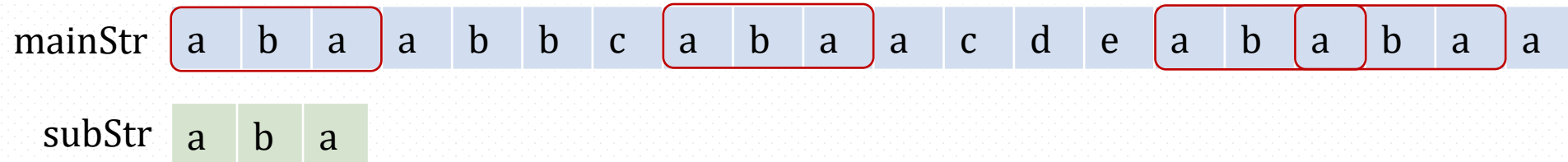
: 메인 문자열의 모든 substring을 검색 하면 시간초과 => 검색 최적화 필요

검색 최적화 keyword

- subStr과 “일치”하는 substring 개수 파악
- subStr의 길이 “최대 4”



hash?



해법 연구

getCount(char subStr[]) 처리

- Hash

rev = 0

mainStr a b a a

str	cnt
a	3
b	1
ab	1
ba	1
aa	1
aba	1
baa	1
abaa	1

해법 연구

getCount(char subStr[]) 처리

- Hash

rev = 0

mainStr a b a a **b**

“b” , “ab” , “aab” , “baab”

str	cnt
a	3
b	2
ab	2
ba	1
aa	1
aba	1
baa	1
abaa	1
aab	1
baab	1

해법 연구

getCount(char subStr[]) 처리

- Hash

rev = 1

mainStr a b a a

str	cnt
a	3
b	1
ab	1
ba	1
aa	1
aba	1
baa	1
abaa	1

해법 연구

getCount(char subStr[]) 처리

- Hash

rev = 1

mainStr **b** a b a a

“b” , “ba” , “bab” , “baba”

str	cnt
a	3
b	2
ab	1
ba	2
aa	1
aba	1
baa	1
abaa	1
bab	1
baba	1

해법 연구

getCount(char subStr[]) 처리

- DA(direct address table)

- subStr로 가능한 모든 상태 수 : $27^4 - 1 = 531,440$
: 개수를 기록하는 int cnt[27^4]
- a=1, z=26 으로 27진법 표현하여 index 사용
ab = $1*27 + 2$
bca = $2*27^2 + 3*27 + 1$

531,440

str	idx	cnt
a	1	10
b	2	6
..		
ab	29	5
..		
abc	786	4
..		

mainStr

a	b	a	a	b	b	c	a	b	a	a	c	d	e	a	b	a	b	a	a
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

해법 연구

자료 구성

- **mainStr 관리** : 앞뒤로 삽입/삭제

- 1) container : list or deque (C++) / deque (python)

- 2) 배열을 (총 길이 * 2)로 생성하여 중간부터 사용

- **4자리 이하의 substring 개수 관리**

- 1) hash table : unordered_map<string, int> htab (C++) / dict (python)

- 2) direct address table : int cnt[27*27*27*27]

감사합니다