

알고리즘 스터디

2주차: 자료구조와 문자열 처리



자료구조

➤ 데이터를 구성하고 저장하는 방법

- 어떤 방식으로 읽고 쓰지에 따라 사용하는 자료구조가 달라야함

➤ 자료구조의 종류

- 선형 자료구조 vs 비선형 자료구조

➤ 선형 자료구조

- 배열, 링크드 리스트, 스택, 큐

➤ 비선형 자료구조

- 트리, 그래프



선형자료구조: 스택(Stack)

- 마지막에 들어간 자료가 먼저 나오는 LIFO(Last-In-First-Out)
- 가능한 연산
 - PUSH: 스택 맨 위에 자료를 쌓는다
 - POP: 스택 맨 위에 있는 자료에 접근하고 제거한다
- CPP 기준 <stack.h> 에 정의되어있다

선형자료구조: 큐(Queue)

- 스택과 큐의 기능을 동시에 제공하는 자료구조
- 가능한 연산
 - Enqueue: 스택 맨 뒤에 자료를 넣는다
 - Dequeue: 스택 맨 앞에서 자료에 접근하고 제거한다
- CPP 기준 <queue.h> 에 정의되어있다

선형자료구조: 덱(Deque)

- Double-Ended Queue
- 양쪽 끝에서 삽입과 삭제가 가능한 자료구조
- 가능한 연산
 - PushBack: 덱의 맨 뒤에 자료를 넣는다
 - PushFront: 덱의 맨 앞에 자료를 넣는다
 - PopBack: 덱의 맨 뒤에서 자료에 접근하고 제거한다
 - PopFront: 덱의 맨 앞에서 자료에 접근하고 제거한다
- CPP 기준 <deque.h> 에 정의되어있다

문자열

➤ 문자로 이루어진 배열

➤ 관련 알고리즘

- 단순 조작 (','를 기준으로 자르기, YYYY-MM-DD형태의 날짜 비교하기)
- 패턴 매칭 (LCS, 문자열 S내 패턴 P가 있는지 찾기)
- 문자열 정렬, 압축 등...

문자열: 예제

- 날짜를 나타내는 아래 두 문자열의 날짜 차이를 구해보자
 - “1999-02-26”
 - “2018-03-01”

문자열: C++

```
1  #include <sstream>
2  #include <string>
3  #include <iostream>
4  #include <vector>
5
6  using namespace std;
7
8  ► int main() {
9      string s;
10     vector<string> first_date, second_date;
11
12     cout << "첫번째 날짜를 YYYY-MM-DD 형식으로 입력" << '\n';
13     cin >> s;
14     stringstream ss1(s);
15     while (getline( &: ss1, &: s, dlm: '-'))
16         first_date.push_back(s);
17
18     cout << "두번째 날짜를 YYYY-MM-DD 형식으로 입력" << '\n';
19     cin >> s;
20     stringstream ss2(s);
21     while (getline( &: ss2, &: s, dlm: '-'))
22         second_date.push_back(s);
23
24     int days_1 = stoi( str: first_date[0]) * 12 * 28 + stoi( str: first_date[1]) * 28 + stoi( str: first_date[2]);
25     int days_2 = stoi( str: second_date[0]) * 12 * 28 + stoi( str: second_date[1]) * 28 + stoi( str: second_date[2]);
26
27     cout << "두 날짜는 " << abs(days_2 - days_1) << "일 만큼 차이 납니다" << '\n';
28
29     return 0;
30 }
```


문자열: Python

```
1  from datetime import datetime
2
3
4  1개의 사용 위치
5  ▼ def main():
6      first_date = input("첫번째 날짜를 입력하세요. (YYYY-MM-DD) : ")
7      second_date = input("두번째 날짜를 입력하세요. (YYYY-MM-DD) :")
8      first_date = datetime.strptime(first_date, "%Y-%m-%d")
9      second_date = datetime.strptime(second_date, "%Y-%m-%d")
10     print(abs((first_date - second_date).days))
11
12  ► if __name__ == '__main__':
13     main()
```

연습문제: KMP는 왜 KMP일까?[1]

KMP 알고리즘이 KMP인 이유는 이를 만든 사람의 성이 Knuth, Morris, Prett이기 때문이다. 이렇게 알고리즘에는 발견한 사람의 성을 따서 이름을 붙이는 경우가 많다. 또 다른 예로, 유명한 비대칭 암호화 알고리즘 RSA는 이를 만든 사람의 이름이 Rivest, Shamir, Adleman이다. 사람들은 이렇게 사람 성이 들어간 알고리즘을 두 가지 형태로 부른다.

- 첫 번째는 성을 모두 쓰고, 이를 하이픈(-)으로 이어 붙인 것이다. 예를 들면, Knuth-Morris-Pratt이다. 이것을 긴 형태라고 부른다.
- 두 번째로 짧은 형태는 만든 사람의 성의 첫 글자만 따서 부르는 것이다. 예를 들면, KMP이다.

[1] <https://www.acmicpc.net/problem/2902>

연습문제: KMP는 왜 KMP일까?[1]

동혁이는 매일매일 자신이 한 일을 모두 메모장에 적어놓는다. 잠을 자기 전에, 오늘 하루 무엇을 했는지 되새겨 보는 것으로 하루를 마감한다. 하루는 이 메모를 보던 중, 지금까지 긴 형태와 짧은 형태를 섞어서 적어 놓은 것을 발견했다.

이렇게 긴 형태로 하루 일을 기록하다가 메모장 가격이 부담되어 파산될 것이 뻔하기 때문에, 앞으로는 짧은 형태로 기록하려고 한다. 긴 형태의 알고리즘 이름이 주어졌을 때, 이를 짧은 형태로 바꾸어 출력하는 프로그램을 작성하시오.

Eg. Knuth-Morris-Pratt -> KMP

[1] <https://www.acmicpc.net/problem/2902>