2학년 1학기 주제연구 발표

주제 : 콘웨이의 둠스데이 알고리즘

문태서

주제 선정 이유

누워서 읽는 알고리즘

퍼즐로 풀어보는 알고리즘의 세계

데이터 구조 정의하기

P를 출력하는 프로그램 P

숨어 있는 버그를 찾아라

톡톡 튀는 알고리즘의 세계

펠린드롬의 세계

콘웨이의 둠스데이 알고리즘

정렬 알고리즘

검색 알고리즘과 최적화 문제

동적 프로그래밍

해시 알고리즘

사운덱스 검색 알고리즘

수도사 메르센느

프로그래머가 느끼는 성취감의 본질

문학적 프로그래밍

유클리드 알고리즘

재귀의 마술

RSA 알고리즘

잠깐 쉬어 가기

RSA 알고리즘 - 계속

세 줄짜리 펄 프로그램

해커들이 작성한 시 이해하기

두 줄짜리 RSA 알고리즘

N개의 여왕 문제

눈으로 풀어 보는 N개의 여왕 문제

문제 속에 숨어 있는 단편적인 알고리즘

재귀와 스택

제프 소머즈의 알고리즘

비트 연산자 복습하기

2의 보수

제프 소머즈 알고리즘 분석

둠스데이 알고리즘

특정한 날짜의 요일을 찾아내는 알고리즘

2월의 마지막 날 (둠스데이)을 구하기 위해 한 해마다 둠스데이의 요일이 증가하는 규칙을 활용한다.

한 해의 2월의 마지막날 (둠스데이)의 요일을 알면, 그 날과 요일이 무조건 같은 날들을 알 수 있다.

구하고자 하는 달에서 둠스데이와 요일이 같은 날짜(2번 특징)를 토대로 특정일의 요일을 계산할 수 있다.

+ 윤년에 대한 개념

윤년 : 2월의 마지막 날이 28일이 아닌 29일이 되는 해

년도가 4의 배수인 해는 윤년이라고 이름을 부르고, 일년이 366일 이 된다.

그러나 만약, 년도가 100의 배수인데 400으로 나누어떨어지지 않으면 그 해는 평년으로 한다.

-> 년도가 4로 나누어지면서, 400으로 나누어떨어지는 해는 윤년이다.

ex) year % 4 == 0 && year % 100 != 0 || year % 400 == 0) ? "윤년" : "평년"

1번 특징.

구하고자 하는 해의 2월의 마지막 날을 “둠스데이"로 칭한다.

특정 년도의 둠스데이를 구하는 방법은 2가지가 있다.

1. 가까운 년도에 대해서 사용할 만한 방법

2. 모든 년도에 대해서 적용 가능한 방법

1번 규칙 - 방법 첫 번째.

가까운 년도에 대해서 사용할만한 방법

둠스데이는 매년 한 요일씩 증가한다.

-> 2021년의 둠스데이가 “일요일" 이었으므로, 2022년은 “월요일" 이고,
2023년은 “화요일” 이다.

예외적으로 윤년으로 넘어갈 때는 둠스데이가 2일 증가한다.

-> 2024년은 윤년이므로 “목요일" 이 된다.

연도별 둠스데이에 해당하는 요일

월	화	수	목	금	토	일	월	화	수	목	금	토	일
1898	1899	1900	1901	1902	1903	→	1904	1905	1906	1907	→	1908	1909
1910	1911	→	1912	1913	1914	1915	→	1916	1917	1918	1919	→	1920
1921	1922	1923	→	1924	1925	1926	1927	→	1928	1929	1930	1931	→
1932	1933	1934	1935	→	1936	1937	1938	1939	→	1940	1941	1942	1943
→	1944	1945	1946	1947	→	1948	1949	1950	1951	→	1952	1953	1954
1955	→	1956	1957	1958	1959	→	1960	1961	1962	1963	→	1964	1965
1966	1967	→	1968	1969	1970	1971	→	1972	1973	1974	1975	→	1976
1977	1978	1979	→	1980	1981	1982	1983	→	1984	1985	1986	1987	→
1988	1989	1990	1991	→	1992	1993	1994	1995	→	1996	1997	1998	1999
→	2000	2001	2002	2003	→	2004	2005	2006	2007	→	2008	2009	2010
2011	→	2012	2013	2014	2015	→	2016	2017	2018	2019	→	2020	2021
2022	2023	→	2024	2025	2026	2027	→	2028	2029	2030	2031	→	2032
2033	2034	2035	→	2036	2037	2038	2039	→	2040	2041	2042	2043	→
2044	2045	2046	2047	→	2048	2049	2050	2051	→	2052	2053	2054	2055
→	2056	2057	2058	2059	→	2060	2061	2062	2063	→	2064	2065	2066
2067	→	2068	2069	2070	2071	→	2072	2073	2074	2075	→	2076	2077
2078	2079	→	2080	2081	2082	2083	→	2084	2085	2086	2087	→	2088
2089	2090	2091	→	2092	2093	2094	2095	→	2096	2097	2098	2099	2100

1번 규칙 - 방법 두 번째.

모든 년도에 대해서 적용 가능한 방법

[1] 세기별로 기준일이 있다.

1800 ~ 1899 : 금요일
1900 ~ 1999 : 수요일
2000 ~ 2099 : 수요일
2100 ~ 2199 : 일요일

금, 수, 화 일의 기준일은 계속해서 반복된다. (ex. 2200 ~ 2299 : 금요일)

[2]

A : 년도의 끝 두자리 수를 12로 나눈 몫
B : 년도의 끝 두 자리 수를 12로 나는 나머지
C : B를 4로 나눈 몫
→ 기준일 + A + B + C = “둠스데이"

계산 예시. (둠스데이)

2021년을 예시로 계산해보면

2021년 기준일 : 화요일

A : 21 // 12 = 1

B : 21 mod12 = 9

C : 9 // 4 = 2

-> 화요일 + 12 = 일요일 임을 확인할 수 있다.

2번 특징. (짝수 달)

각 달마다 둠스데이의 요일과 같은 요일의 날짜를 구할 수 있다.

2월의 마지막 날 = “둠스데이"
윤년 X -> 29일
윤년 O -> 28일

4월 4일

6월 6일

8월 8일

10월 10일

12월 12일

2번 특징. (홀수 달)

각 달마다 둠스데이의 요일과 같은 요일의 날짜를 구할 수 있다.

1월
- 윤년일 경우 1월 4일
- 윤년이 아닐 경우 1월 3일

3월
- 3월 21일

5월 / 9월
- 5월 9일
- 9월 5일

7월 / 11월
- 7월 11일
- 11월 7일

계산 예시. (전체)

2021년 크리스마스를 예시로 계산해보자 (12월 25일)

1. 2021년의 둠스데이는 일요일
2. 12월의 둠스데이와 요일이 같은 날은 12일
3. 25일 - 12일 = 13일
4. 13 mod 7 = 6이므로 일요일에서 6일 지난 토요일이 정답이 된다.
   (A mod B 의 값은 A를 B로 나눈 나머지이다.)

-> 2021년 크리스마스의 요일은 “토요일" 이다.

구현 코드

```python
# 이 파일은 콘웨이의 둠스데이 알고리즘을 파이썬으로 구현한 소스코드입니다. - 선린 주제연구 발표
import sys
y, m, d = map(int, sys.stdin.readline().rstrip().split()) # 입력 포멧 : (년도 4자리) (월 2자리) (일 2자리)

dic = {'1': 0, '2': 0, '3': 21, '4': 4, '5': 9, '6': 6, '7': 11, '8': 8, '10': 10, '11': 7, '9': 5, '12': 12
dic2 = {1: "월요일", 2: "화요일", 3: "수요일", 4: "목요일", 5: "금요일", 6: "토요일", 7: "일요일", 0: "일요일"}

if (y % 4 == 0) and (y % 100 == 0): # 윤년
    dic['1'] = 4; dic['2'] = 29
else: # 평년
    dic['1'] = 3; dic['2'] = 28

if y >= 2100:
    std = 7
elif y >= 2000:
    std = 2
elif y >= 1900:
    std = 3
elif y >= 1800:
    std = 5
dooms = (std + ((y % 100) // 12) + ((y % 100) % 12) + ((y % 100) % 12 // 4)) % 7 # (기준일 + A + B + C) % 7

same_d = dic.get(str(m))

Day = d - same_d

print(f'{y}년 {m}월 {d}일은 {dic2.get((dooms + Day) % 7)} 입니다.')
```

발표 끝.