

# 알고리즘

- 3주차: 재귀(Recursion) -



데이터네트워크연구실  
정진욱

[jinwook1240@cs-cnu.org](mailto:jinwook1240@cs-cnu.org)

# Standard Library

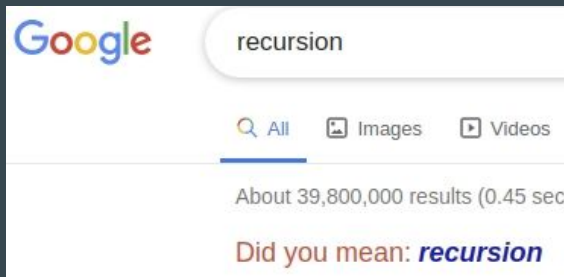
- Python 3: <https://docs.python.org/3/library/>
- JAVA 8: <https://cr.openjdk.java.net/~iris/se/11/latestSpec/api/>
- C: <https://en.cppreference.com/w/c>
- CPP: <https://en.cppreference.com/w/cpp>

ID/ext	external ID	name
adb	ada	Ada
awk	awk	AWK
bash	bash	Bash shell
c	c	C
csharp	csharp	C#
cpp	cpp	C++
f95	f95	Fortran
hs	haskell	Haskell
java	java	Java
js	javascript	JavaScript
kt	kotlin	Kotlin
lua	lua	Lua
pas	pascal	Pascal
pl	pl	Perl
sh	sh	POSIX shell
plg	prolog	Prolog
py2	python2	Python 2
py3	python3	Python 3
r	r	R
rb	ruby	Ruby
scala	scala	Scala
swift	swift	Swift

오늘의 주제: 재귀 (Recursion)

# 재귀 (Recursion)

- Recursion in computer science is a method of solving a problem where the solution depends on solutions to smaller instances of the same problem (as opposed to iteration)



어느 한 컴퓨터공학과 학생이 유명한 교수님을 찾아가 물었다.

"재귀함수가 뭔가요?"

"잘 들어보게. 옛날에 산 꼭대기에 현자가 있었어. 질문엔 모두 지혜롭게 대답해 주었지.

그런데 어느날, 그 선인에게 한 선비가 찾아와서 물었어.

"재귀함수가 뭔가요?"

"잘 들어보게. 옛날에 산 꼭대기..."

# 재귀 가산점

- 이번 목표 : 재귀
  - 그러나 문제 자체는 반복으로 해결 가능
  - 그래서, 재귀로 해결 시 가산점(1점)
- 가산점 받는 법
  - 코딩테스트에 재귀를 이용한 정답 제출
  - 보고서에 재귀를 이용하였다고 기재
- 재귀 에러 방지
  - python 재귀 횟수 제한이 있음

```
RecursionError: maximum recursion depth exceeded in comparison
```

- 해결 방법 : 아래 코드를 실행하면 재귀 횟수 제한이 30000번으로 늘어남

```
import sys
sys.setrecursionlimit(30000)
```

문제1(출석) [1]: 토끼 수

# 토끼 수

- 첫 달에는 새로 태어난 토끼 한 쌍만이 존재한다.  
두 달 이상 지난 토끼 한 쌍은 매달 새끼 한 마리를 낳는다.  
토끼는 죽지 않는다.  
n번째 달의 토끼 수는?
- 첫 달 토끼 수 2마리, 2번째 달 토끼 수 2마리
- Input  
n # n번째 토끼 수 구하기( $n < 150$ )
- Output  
(토끼 수)

예시 입력

11

예시 출력

30

예시 입력

40

예시 출력

252734

예시 입력

30

예시 출력

11171

문제2 [2][+1]: 징검다리



# 징검다리

- 가위로 이기면  $x$ 칸, 바위로 이기면  $y$ 칸, 보로 이기면  $z$ 칸 앞으로 이동한다.  
두 사람 간의 거리가  $n$ 일 때, 두 사람이 가장 빨리 만나는 경우의 가위바위보 횟수를 구하여라. 만날 수 없을 경우 -1 출력, 이기는 경우만 구할 것

- Input:

$x\ y\ z$       #  $x$ 칸,  $y$ 칸,  $z$ 칸( $x,y,z < 30$ )  
 $n$             # 두 사람의 거리( $n < 10000$ )

- Output:

$a$             # 최소 가위바위보 횟수

예시 입력

1 2 5  
3

예시 출력

2

예시 입력

3 5 22  
80

예시 출력

7

예시 입력

8 4 12  
19

예시 출력

-1

문제3 [2][+1]: 회문 사전

# 회문(Palindrome)

- 거꾸로 읽어도 똑같이 읽히는 문자열  
(<https://ko.wikipedia.org/wiki/%ED%9A%8C%EB%AC%B8>)
- 예시
  - 기러기, 요기요
  - 나갔다오나 나오다갔나
  - race car, nurses run

다 시 합 창 합 시 다  
시 합 창 합 시  
합 창 합  
창

# 회문 사전(재귀)

- 길이는  $n$ 이고,  $m$ 종류의 문자로 이루어진 단어 사전 (ex : 5종류 -abcde)  
문자는 모두 소문자

- Input

$n$   $m$  # 정수( $n < 100$ ,  $m < 26$ )

$x$  #  $x$ 번째 문자열 출력하기  
( $0 < x < 100000$ )

- Output

( $x$ 번째 문자열)

예시 입력

100 26  
26

예시 출력

aaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaazzaaaaaa  
aaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaaaaaa  
aaaa

11 4  
5

aaaaaaaaaaaa  
aaaaabaaaaa  
aaaaacaaaaa  
aaaaadaaaaa  
aaaababaaaa  
aaaabbbaaaa  
aaaabcbaaaa  
aaaabdbaaaa  
aaaacacaaaa  
aaaacbcaaaa  
aaaacccaaaa

...

ddddddddddddd

예시 입력

11 4  
5

예시 출력

aaaababaaaa

예시 입력

5 26  
549

예시 출력

avcva

예시 입력

30 25  
56832

예시 출력

aaaaaaaaaaaadpxggxpdaaaaaa  
aaaaa

# 언어별 문자열 다루는 방법

- Python3
  - string 연산(인덱싱 가능)
- Java
  - StringBuilder
    - <https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/StringBuilder.html>
  - 혹은 String 연산
- C++
  - char 배열 이용

# 아스키 코드표

DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII	DEC	ASCII
1	☺	32	space	64	@	96	`	128	Ç	160	á	192	Ł	224	Ó
2	☹	33	!	65	A	97	a	129	ü	161	í	193	ł	225	ô
3	♥	34	"	66	B	98	b	130	è	162	ó	194	ṽ	226	õ
4	♦	35	#	67	C	99	c	131	â	163	ú	195	ṽ	227	ö
5	♣	36	\$	68	D	100	d	132	ä	164	ñ	196	—	228	ø
6	♠	37	%	69	E	101	e	133	à	165	Ñ	197	+	229	Ô
7	•	38	&	70	F	102	f	134	å	166	ª	198	ä	230	µ
8	▣	39	'	71	G	103	g	135	ç	167	º	199	Ä	231	þ
9	○	40	(	72	H	104	h	136	ê	168	¿	200	ℓ	232	ð
10	▣	41	)	73	I	105	i	137	ë	169	®	201	℥	233	Ú
11	♂	42	*	74	J	106	j	138	è	170	¬	202	℥	234	Ù
12	♀	43	+	75	K	107	k	139	ï	171	½	203	℥	235	Û
13	♪	44	,	76	L	108	l	140	î	172	¼	204	℥	236	ý
14	♪	45	-	77	M	109	m	141	ì	173	í	205	≡	237	ÿ
15	☼	46	.	78	N	110	n	142	Ä	174	«	206	≡	238	˘
16	▶	47	/	79	O	111	o	143	Å	175	»	207	≡	239	˙
17	◀	48	0	80	P	112	p	144	È	176	⌘	208	ð	240	˚
18	↕	49	1	81	Q	113	q	145	æ	177	⌘	209	Ð	241	±
19	!!	50	2	82	R	114	r	146	Æ	178	⌘	210	Ê	242	≡
20	¶	51	3	83	S	115	s	147	ô	179		211	Ë	243	¾
21	§	52	4	84	T	116	t	148	ö	180	†	212	Ë	244	¶
22	—	53	5	85	U	117	u	149	ò	181	Á	213	ı	245	§
23	↕	54	6	86	V	118	v	150	û	182	Â	214	í	246	÷
24	↕	55	7	87	W	119	w	151	ù	183	À	215	î	247	˘
25	↕	56	8	88	X	120	x	152	ÿ	184	©	216	ï	248	˚
26	→	57	9	89	Y	121	y	153	Ö	185	≡	217	ı	249	˙
27	←	58	:	90	Z	122	z	154	Ü	186	≡	218	ı	250	˙
28	└	59	;	91	[	123	{	155	ø	187	≡	219	ı	251	ı
29	↕	60	<	92	\	124		156	£	188	≡	220	ı	252	ı
30	▲	61	=	93	]	125	}	157	Ø	189	€	221	ı	253	ı
31	▼	62	>	94	^	126	~	158	×	190	¥	222	ı	254	ı
		63	?	95	_	127	△	159	f	191	ı	223	ı	255	space

문제4 [2][+1]: 3가지 괄호

# 3가지 괄호

- 프로그램에서 쓰이는 괄호는 3가지가 있다.  
[], {}, ()인데, 세 가지 괄호를 각각 l,m,n개 배치할 때,  
배치 규칙에 맞는 경우는 몇 가지인지 구하여라

- Input :  
l m n # 세 가지 숫자(l,m,n<30)
- Output :  
(괄호 배치하는 경우의 수)

- 배치 규칙 :  
가능 - (( )), ({}), ()[], ...  
불가능 - ({}), [[]], (}

예시 입력
1 1 1
예시 출력
30

예시 입력
3 3 3
예시 출력
8168160

[](){}  
[]{}()  
... -6개  
[{}]()  
... -6개  
()[{}]  
... -6개  
({})[]  
... -6개  
({})()  
... - 6개



# 참고
















# 제출 방법: 점수 확인

- <http://coding.cnu.ac.kr:8080/domjudge/public> 에 접근하면 점수가 보임

Algorithm Week01 - Python Basic

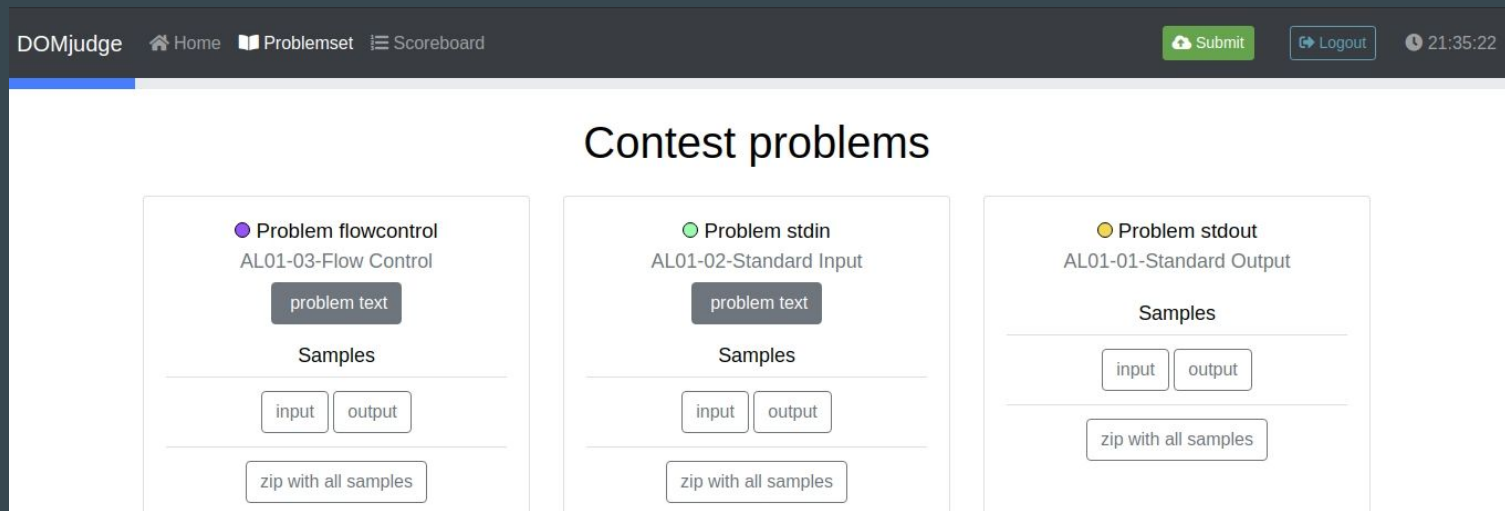
starts: 06:00 - ends: 06:00

Filter ▼

RANK	TEAM	SCORE	FLOWCONTROL  [5 POINTS]	STDIN  [4 POINTS]	STDOUT  [1 POINT]
1	♡ DNLab	0 0			
SUMMARY			 0  0  0  n/a	 0  0  0  n/a	 0  0  0  n/a

# 제출 방법: 로그인 후 제출

- 로그인 하면 문제 셋 표시됨: 로그인 계정 및 패스워드는 종이로 나누어줌!
- Submit 을 눌러 코드 제출 (\*.py)
- 점수 표에서 득점 확인 가능



# 제출 방법: ID / PASSWD

- ID / PASSWD 잊어버리면 연락!
- 실습 제출 시스템 ID / PASSWD 변경하고 싶을 경우 메일!

# 잊지 않아야 할 것) 소스코드 및 보고서

- 사이버 캠퍼스에 목요일까지 제출
  - 추가 시간 필요한 학생들도 목요일까지 제출. 추가 시간 문제 해결은 메일로도 제출!
- 보고서(.pdf 파일), 소스코드(.java, .py 등) zip 파일 압축
  - AL\_학번\_이름\_03.zip (메일 추가 제출) or AL19\_03.zip (사이버캠퍼스)
- 시간/공간 복잡도 해석(STL 고려), 자신의 생각, 질문, 느낀점, 공유하고 싶은 문제
  - 문제 해결을 위해 어떤 접근법을 사용하였는지, 무엇을 배웠고 느꼈는가?

# 어디에 제출해야하는가? 헛갈린다.

- 제한 시간 내에 모두 해결했는가?
  - 네 - 사이버캠퍼스에 소스코드와 보고서 압축해서 제출
  - 아니요 -
    - 추가 시간 내에 해결한 문제가 있는가?
      - 네 - 사이버캠퍼스에 소스코드와 보고서 압축해서 제출하고, 메일로도 제출
      - 아니요 - 사이버캠퍼스에 소스코드와 보고서 압축해서 제출

# 보고서 템플릿: .pdf 로 제출!

- 알고리즘-x주차-주제  
학번 이름
- 코드 테스트 결과 (점수표)

1	DOMjudge	0	0						
---	----------	---	---	--	--	--	--	--	--

- 각 문제별 내용
  - a. 문제 / 목표
  - b. 해결 방법 (주요 소스코드 첨부. 스크린샷 or 코드 CV)
  - c. 결과 (입력, 출력 결과)
- 느낀점: 과제를 하며 느낀 점 / 공유하고싶은 문제 / 난이도 / 부탁 / 조교에게... 등

# 3주차 질문

<https://forms.gle/uuckqEM1WAw898EWA>



# 3주차 설문조사

난이도를 조정하고 요청 사항을 받아들이기 위함이니 꼭 참여 바랍니다.

<https://forms.gle/CcFrr1shiSJ6RFjWA>