

# 제1회 숙명여자대학교 교내 알고리즘 경진 대회 **SMUPC**

Official Solutions



*with*

ALGOS



DSC Sookmyung



문제	의도한 난이도	출제자
A SMUPC의 등장	Easy	김도은숙명여대 whaeun25
B 눈덩이 굴리기	Medium	김도현홍익대 swoon
C 헌내기는 친구가 필요해	Medium	김도현홍익대 swoon
D SMUPC 계산기	Medium	김도은숙명여대 whaeun25
E 얼음깨기 펭귄	Medium	김도은숙명여대 whaeun25
F 펭귄 네비게이터	Hard	김도현홍익대 swoon
G 도도의 수학놀이	Hard	김도현홍익대 swoon

# A. SMUPC의 등장

출제자 : whaeun25(김도은, Algos[숙명여자대학교])

가장 먼저 푼 사람 : 00:04 yoonyesle

정답률 : 44.3% ( 90정답 203제출 )

## A. SMUPC의 등장

a	97	h	104	o	111	v	118
b	98	i	105	p	112	w	119
c	99	j	106	q	113	x	120
d	100	k	107	r	114	y	121
e	101	l	108	s	115	z	122
f	102	m	109	t	116		
g	103	n	110	u	117		

입력으로 주어지는 문자열의 각 문자에 대해  
아스키코드의 각자리 숫자의 합만큼 반복해서  
출력합니다.

# B. 눈덩이 굴리기

출제자 : swoon (김도현, HI-ARC [홍익대학교])

가장 먼저 푼 사람 : 00:18 yoonyesle

정답률 : 7.3% ( 6정답 81제출 )

## B. 눈덩이 굴리기

- 전형적인 완전 탐색 문제입니다.
- dfs or bfs로 완전 탐색을 진행해주면 됩니다.
- $\text{dfs}(i+1, \text{snow}, \text{time}+1), \text{dfs}(i+2, \text{snow} // 2, \text{time}+1)$
- 시간 복잡도 :  $O(2^N)$
- dp를 이용한  $O(N^2)$  풀이도 존재합니다.

# C. 헌내기는 친구가 필요해

출제자 : swoon (김도현, HI-ARC [홍익대학교])

가장 먼저 푼 사람 : 00:17 joki624

정답률 : 20.4% ( 18정답 88제출 )

- 전형적인 BFS 문제입니다.
- 만날 수 있는 사람의 수가 0명인 경우 TT를 출력해야 합니다.
- BFS, DFS, 유니온파인드 등등 원하는 방법으로 푸시면 됩니다.



# D. SMUPC 계산기

출제자 : whaeun25(김도은, Algos[숙명여자대학교])

가장 먼저 푼 사람 : 01:22 hjhsophia1

정답률 : 7.2% ( 6정답 83제출 )

#### D. SMUPC 계산기

알파벳	S	M	U	P	C
해당 수식	-	*	/ (정수 몫 나눗셈)	+	여태까지의 계산 결과값 반환

- 결과값을 저장할 변수를 만들어 계산 결과값을 계속 저장하고 C가 나올때마다 결과값을 출력할 수 있도록 합니다.
- 맨 처음 들어오는 수는 결과값으로 저장하고 알파벳이 들어올 때마다 그에 맞는 계산을 하도록 하여 결과값을 저장하는 변수에 결과값을 저장합니다.
- C가 단 한번도 들어오지 않은 경우에는 **NO OUTPUT**을 출력하도록 합니다.

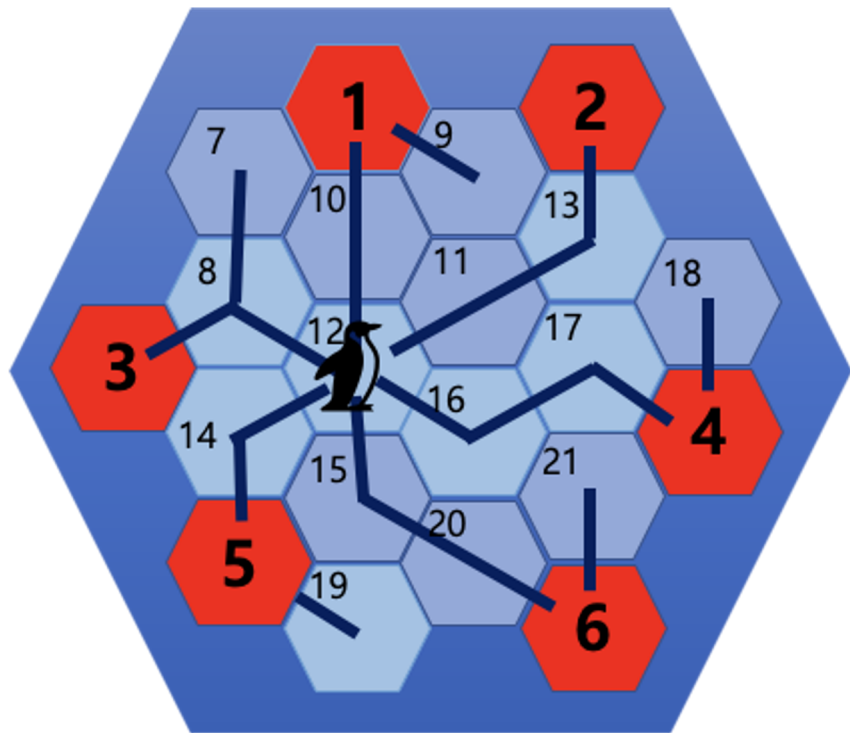
# E. 얼음깨기 펭귄

출제자 : whaeun25(김도은, Algos[숙명여자대학교])

가장 먼저 푼 사람 : 00:00 ???

정답률 : 0% ( 0정답 14제출 )

## E. 얼음깨기 펭귄



### - bfs 탐색 풀이:

펭귄으로부터 지지대까지의 거리를 측정

### - dfs 탐색 풀이:

각 지지대로부터 펭귄까지의 거리를 측정

각 거리 중에서 가장 짧은 거리가 나오는 두 경로를 선택하고 펭귄이 올라가 있는 얼음을 제외한 나머지 얼음들을 모두 깨는 얼음으로 구분하여 구합니다.

# F. 펭귄 네비게이터

출제자 : swoon (김도현, HI-ARC [홍익대학교])

가장 먼저 푼 사람 : 00:00 ???

정답률 : 0% ( 0정답 8제출 )

## [ 조건 ]

- 펭귄은 우측이나 아래로만 이동한다.
- 펭귄은 자신이 이동할 얼음의 숫자가 현재 얼음의 숫자보다 커야 한다.
- 펭귄이 어떤 움직임을 하든지 집에 갈 수 있어야 한다.

위 조건을 바탕으로 문제를 다시 만들어봅시다.

- 현재 얼음의 우측 얼음과 아래 얼음의 숫자는 현재 얼음의 숫자보다 커야 한다.

조건을 무시하고 랜덤한 얼음길을 한 번 만들어봅시다.

어차피 윗 길과 아랫 길은 오름차순 배치가 되어야 하므로 순서에 상관 없이  $2n$ 개의 수 중 위에 들어갈 수  $n$ 개를 뽑은 뒤 정렬하여 위쪽 길에 넣고, 남은  $n$ 개의 수를 정렬하여 아래 길에 넣습니다.

이렇게 만든 얼음길이 조건을 충족하는 얼음길인지 파악하는 방법은 처음에 뽑은  $n$ 개의 숫자에 그보다 큰 숫자를 하나씩 매칭해줄 수 있는지 검사하면 됩니다. 어차피 횡으로는 오름차순으로 정렬이 되었기 때문입니다.

**선택한  $n$ 개의 숫자에 선택하지 않은  $n$ 개의 더 큰 숫자를 각각 매칭하는 방법의 개수가 답이 됩니다.**

1	3	4
2	5	6



( ) ( ( ) )  
1 2 3 4 5 6

선택한  $n$ 개의 수 보다 큰  $n$ 개의 수를 각각 매칭하는 방법은 다르게 생각해 볼 수 있습니다.

- 윗 숫자를 여는 괄호로 본다.
- 밑 숫자를 닫는 괄호로 본다.

이런 방식으로 생각해 안정적인 괄호 문자열이 된다면 매칭이 가능합니다.

이제  $n$ 개의 괄호를 가진 안정적인 괄호 문자열의 개수를 구하는 문제로 바뀌었습니다.



괄호 문자열의 개수를 구하는 방법은 다음과 같습니다.

$C[n]$ 을  $n$ 개의 괄호 쌍을 사용해 괄호 문자열의 개수로 정의합니다.

괄호 문자열에 괄호 쌍 하나를 고정합니다.

그 괄호를 기준으로 괄호 밖과 괄호 안으로 구역을 나눕니다.

괄호 내에 두는 경우의 수 \* 괄호 밖에 두는 경우의 수를 모두 더해줍니다.

따라서 다음과 같은 점화식이 유도됩니다.

$$C[n] = C[0] * C[n-1] + C[1] * C[n-2] + \dots + C[n-1] * C[0]$$

시간복잡도 :  $O(N^2)$

# G. 도도의 수학놀이

출제자 : swoon (김도현, HI-ARC [홍익대학교])

가장 먼저 푼 사람 : 00:00 ???

정답률 : 0% ( 0정답 120제출 )

핵심 1. 수를 먼저 돌린 후 수를 붙여도 된다.

181, 9610, 91 -> 181, 16, 0196 -> 181160196

먼저 수들을 돌려 대소를 비교한 후 수를 붙여도  
수를 붙인 후 돌리는 것과 같습니다.

**핵심 2. 두 수  $a, b$ 가 있을 때,**

**문자열 변환을 통하여 어느 수를 앞에 붙여야 하는지 알 수 있다.**

각 수를 문자열로 바꾼 뒤,  $\text{int}(a+b)$ 와  $\text{int}(b+a)$ 를 비교합니다.

이 방식으로 어느 수가 앞에 오는 것이 최적인지 파악할 수 있습니다.

**핵심 3. 한 번 더 쓸 수 있는 수는 가장 큰 정수를 사용해야 한다.**

예를 들어 돌린 수가 986, 98이 있을 때,

핵심 2 기준에 따라 정렬을 하면 98, 986이 됩니다.

정렬 기준 맨 처음 원소를 두 번 사용하면 98 98 986이 되지만,

가장 큰 정수를 두 번 사용하면 98 986 986으로 더 커질 수 있습니다.

**핵심 4. 만약 수를 추가적으로 한 번 사용했으면 사용한 체크를 해 주어야 한다.**

**체크를 해주지 않는 경우**

**11 1 11의 경우에서 11 11 11 11 1이 나올 수 있습니다.**