

브루트 포스 - 비트마스크

최백준 choi@startlink.io

비트마스크

Bitmask

- 비트(bit) 연산을 사용해서 부분 집합을 표현할 수 있다.

비트 연산

Bitwise operation

- $\&$ (and), $|$ (or), \sim (not), \wedge (xor)

A	B	$\sim A$	$A \& B$	$A B$	$A \wedge B$
0	0	1	0	0	0
0	1	1	0	1	1
1	0	0	0	1	1
1	1	0	1	1	0

비트 연산

Bitwise operation

- 두 수 A와 B를 비트 연산 하는 경우에는 가장 뒤의 자리부터 하나씩 연산을 수행하면 된다.
- A = 27, B = 83인 경우
- $A = 11011_2, B = 1010011_2$
- $A \& B = 19, A \mid B = 91, A \wedge B = 72$

0	0	1	1	0	1	1		0	0	1	1	0	1	1		0	0	1	1	0	1	1	
&	1	0	1	0	0	1	1		1	0	1	0	0	1	1	^	1	0	1	0	0	1	1
-----								-----								-----							
0	0	1	0	0	1	1		1	0	1	1	0	1	1		1	0	0	1	0	0	0	

비트 연산

Bitwise operation

- not 연산의 경우에는 자료형에 따라 결과가 달라진다.
- $A = 83 = 1010011_2$
- $\sim A = 10101100_2$ (8비트 자료형인 경우)
- $\sim A = 11111111\ 11111111\ 11111111\ 10101100_2$ (32비트 자료형인 경우)
- 또, unsigned, signed에 따라서 보여지는 값은 다르다.

비트 연산

Bitwise operation

- shift left (<<) 와 shift right (>>) 연산이 있다.
- $A \ll B$ (A를 왼쪽으로 B비트만큼 민다)
- $1 \ll 0 = 1$
- $1 \ll 1 = 2 \ (10_2)$
- $1 \ll 2 = 4 \ (100_2)$
- $1 \ll 3 = 8 \ (1000_2)$
- $1 \ll 4 = 16 \ (10000_2)$
- $3 \ll 3 = 24 \ (11000_2)$
- $5 \ll 10 = 5120 \ (101000000000000_2)$

비트 연산

7

Bitwise operation

- shift left (<<) 와 shift right (>>) 연산이 있다.
- $A \gg B$ (A를 오른쪽으로 B비트만큼 민다)
- $1 \gg 0 = 1$
- $1 \gg 1 = 0$ (0_2)
- $10 \gg 1 = 5$ (101_2)
- $10 \gg 2 = 2$ (10_2)
- $10 \gg 3 = 1$ (1_2)
- $30 \gg 1 = 15$ (1111_2)
- $1024 \gg 10 = 1$ (1_2)

비트 연산

Bitwise operation

- $A \ll B$ 는 $A \times 2^B$ 와 같다.
- $A \gg B$ 는 $A / 2^B$ 와 같다.
- $(A + B) / 2$ 는 $(A+B) \gg 1$ 로 쓸 수 있다.

비트마스크

Bitmask

- 정수로 집합을 나타낼 수 있다.
- $\{1, 3, 4, 5, 9\} = 570 = 2^1 + 2^3 + 2^4 + 2^5 + 2^9$

비트마스크

Bitmask

- 정수로 집합을 나타낼 수 있다.
- $\{1, 3, 4, 5, 9\} = 570 = 2^1 + 2^3 + 2^4 + 2^5 + 2^9$

10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	1	1	1	0	1	0

비트마스크

Bitmask

- 보통 0부터 $N-1$ 까지 정수로 이루어진 집합을 나타낼 때 사용한다.
- 1부터 N 까지 정수로 이루어진 집합을 사용하는 건 공간이 2배 더 필요하다.
- 또, 각종 연산을 조금 변형해서 사용해야 한다.
- 따라서, 0부터 $N-1$ 까지로 변형해서 사용하는 것이 더 좋다.

비트마스크

Bitmask

- $\{1, 3, 4, 5, 9\} = 570$
- 0이 포함되어 있는지 검사
 - $570 \ \& \ 2^0 = 570 \ \& \ (1 \ll 0) = 0$
- 1이 포함되어 있는지 검사
 - $570 \ \& \ 2^1 = 570 \ \& \ (1 \ll 1) = 2$
- 2이 포함되어 있는지 검사
 - $570 \ \& \ 2^2 = 570 \ \& \ (1 \ll 2) = 0$
- 3이 포함되어 있는지 검사
 - $570 \ \& \ 2^3 = 570 \ \& \ (1 \ll 3) = 8$

```

      1000111010
    & 0000000100
    -----
      0000000000

      1000111010
    & 0000001000
    -----
      0000001000
  
```

비트마스크

Bitmask

• {1, 3, 4, 5, 9} = 570

• 1 추가하기

• $570 \mid 2^1 = 570 \mid (1 \ll 1) = 570 \text{ (1000111010}_2\text{)}$

1000111010

| 00000000100

1000111110

• 2 추가하기

• $570 \mid 2^2 = 570 \mid (1 \ll 2) = 574 \text{ (1000111110}_2\text{)}$

1000111010

| 00000001000

1000111010

• 3 추가하기

• $574 \mid 2^3 = 570 \mid (1 \ll 3) = 570 \text{ (1000111010}_2\text{)}$

• 4 추가하기

• $574 \mid 2^4 = 570 \mid (1 \ll 4) = 570 \text{ (1000111010}_2\text{)}$

비트마스크

Bitmask

• {1, 3, 4, 5, 9} = 570

• 1 제거하기

• $570 \ \& \sim 2^1 = 570 \ \& \sim (1 \ll 1) = 568 \ (1000111000_2)$

```
1000111010
& 1111110111
-----
```

• 2 제거하기

• $570 \ \& \sim 2^2 = 570 \ \& \sim (1 \ll 2) = 570 \ (1000111010_2)$

```
1000111010
```

• 3 제거하기

• $562 \ \& \sim 2^3 = 562 \ \& \sim (1 \ll 3) = 562 \ (1000110010_2)$

```
1000111010
| 1111111011
-----
```

• 4 제거하기

• $562 \ \& \sim 2^4 = 562 \ \& \sim (1 \ll 4) = 546 \ (1000101010_2)$

```
1000111010
```

비트마스크

Bitmask

• {1, 3, 4, 5, 9} = 570

• 1 토글하기

• $570 \wedge 2^1 = 570 \wedge (1 \ll 1) = 568 \text{ (1000111000}_2\text{)}$

$$\begin{array}{r} 1000111010 \\ \wedge 0000000100 \\ \hline \end{array}$$

• 2 토글하기

• $570 \wedge 2^2 = 570 \wedge (1 \ll 2) = 574 \text{ (1000111110}_2\text{)}$

$$\begin{array}{r} 1000111010 \\ \wedge 0000001100 \\ \hline 1000111110 \end{array}$$

• 3 토글하기

• $574 \wedge 2^3 = 570 \wedge (1 \ll 3) = 562 \text{ (1000110010}_2\text{)}$

$$\begin{array}{r} 1000111110 \\ \wedge 0000100100 \\ \hline \end{array}$$

• 4 추가하기

• $574 \wedge 2^4 = 570 \wedge (1 \ll 4) = 554 \text{ (1000101010}_2\text{)}$

$$\begin{array}{r} 1000110010 \\ \wedge 0001000000 \\ \hline \end{array}$$

비트마스크

Bitmask

- 전체 집합
 - $(1 \ll N) - 1$
- 공집합
 - 0

비트마스크

Bitmask

- 현재 집합이 S일때
- i를 추가
 - $S \mid (1 \ll i)$
- i를 검사
 - $S \& (1 \ll i)$
- i를 제거
 - $S \& \sim(1 \ll i)$
- i를 토글 (0을 1로, 1을 0으로)
 - $S \wedge (1 \ll i)$

비트 연산

Bitwise operation

- 비트 연산을 사용할 때는 연산자 우선 순위를 생각해야 한다.
- $1 \ll N - 1$ 은 $(1 \ll N) - 1$ 일까? $1 \ll (N - 1)$ 일까?

비트 연산

Bitwise operation

- 비트 연산을 사용할 때는 연산자 우선 순위를 생각해야 한다.
- $1 \ll N - 1$ 은 $(1 \ll N) - 1$ 일까? $1 \ll (N - 1)$ 일까?
- 정답은 $1 \ll (N - 1)$

집합

20

<https://www.acmicpc.net/problem/11723>

- 비트마스크를 연습해보는 문제

집합

<https://www.acmicpc.net/problem/11723>

- 소스: <http://codeplus.codes/6e1c2ed60f20464bb32ad0b98e53f031>

비트마스크

Bitmask

- 물론 배열을 사용하는 것이 더욱 편리하지만, 비트마스크를 사용하는 이유는 집합을 배열의 인덱스로 표현할 수 있기 때문이다.
- 상태 다이내믹을 할 때 자주 사용하게 된다.

bitset

bitset

- C++ 기준으로 int는 32비트, long long는 64비트이다.
- 64비트를 넘는 비트는 정수로 나타낼 수 없다.
- 이런 경우에는 C++은 `bitset`을 이용하면 된다.

부분집합의 합

<https://www.acmicpc.net/problem/1182>

- 서로 다른 N개의 정수로 이루어진 집합이 있을 때, 이 집합의 공집합이 아닌 부분집합 중에서 그 집합의 원소를 다 더한 값이 S가 되는 경우의 수를 구하는 문제
- $1 \leq N \leq 20$

부분집합의 합

25

<https://www.acmicpc.net/problem/1182>

- 모든 집합의 개수 = 2^N
- 모든 집합을 구해보면 된다!

부분집합의 합

<https://www.acmicpc.net/problem/1182>

- 전체 집합 = $(1 \ll N) - 1$

```
for (int i=0; i<(1<<n); i++) {  
  
}
```

부분집합의 합

<https://www.acmicpc.net/problem/1182>

- 전체 집합 = $(1 \ll N) - 1$
- 공집합은 제외해야 한다

```
for (int i=1; i<(1<<n); i++) {  
  
}
```

부분집합의 합

<https://www.acmicpc.net/problem/1182>

- 전체 집합 = $(1 \ll N) - 1$
- 공집합은 제외해야 한다
- 집합에 무엇이 포함되어 있는지 확인하기

```
for (int i=1; i<(1<<n); i++) {  
    for (int k=0; k<n; k++) {  
        if (i&(1<<k)) {  
            }  
        }  
    }  
}
```

부분집합의 합

<https://www.acmicpc.net/problem/1182>

```
for (int i=1; i<(1<<n); i++) {  
    int sum = 0;  
    for (int k=0; k<n; k++) {  
        if (i&(1<<k)) {  
            sum += a[k];  
        }  
    }  
    if (sum == s) {  
        ans += 1;  
    }  
}
```

부분집합의 합

30

<https://www.acmicpc.net/problem/1182>

- 소스: <http://codeplus.codes/1b9e73111817405697121eee82e1bc1b>

스타트와 링크

<https://www.acmicpc.net/problem/14889>

- N명을 N/2명씩 두 팀으로 나누려고 한다. ($4 \leq N \leq 20$, N은 짝수)
- 두 팀의 능력치를 구한 다음, 차이의 최소값을 구하는 문제
- $S[i][j] = i$ 번 사람과 j 번 사람이 같은 팀에 속했을 때, 팀에 더해지는 능력치
- 팀의 능력치: 팀에 속한 모든 쌍의 $S[i][j]$ 의 합

스타트와 링크

<https://www.acmicpc.net/problem/14889>

- 각 사람을 두 팀 중 하나로 나누는 문제이기 때문에, 비트마스크를 이용할 수 있다.
- 비트가 0인 사람은 0번 팀, 1인 사람은 1번 팀이라고 하고
- 전체 경우의 수를 순회할 수 있다.

스타트와 링크

<https://www.acmicpc.net/problem/14889>

- 소스 1: <http://codeplus.codes/d58dfc29edac47ee81920cab06ec7ed1>
- 소스 2: <http://codeplus.codes/0016875c0b8848e9bf6a68a0048f7cff>

종이 조각

<https://www.acmicpc.net/problem/14391>

- $N \times M$ 크기의 종이를 조각으로 잘라서 합의 최대값을 구하는 문제 ($1 \leq N, M \leq 4$)

4	9	3	7
2	5	9	1
3	8	4	6
9	1	5	0

종이 조각

<https://www.acmicpc.net/problem/14391>

- 종이를 조각으로 잘라서 합의 최대값을 구하는 문제

4 9 3			7
2	5	9	1
3	8	4	6
9	1	5	0

종이 조각

<https://www.acmicpc.net/problem/14391>

- 어차피 3자리 수는 항상 4자리 수보다 작다
- 정답은 네자리 수 4개 합이 아닐까?

종이 조각

<https://www.acmicpc.net/problem/14391>

- 어차피 3자리 수는 항상 4자리 수보다 작다
- 정답은 네자리 수 4개 합이 아닐까?

0	0	0	5
0	0	0	0
0	0	0	0
5	0	0	0

종이 조각

<https://www.acmicpc.net/problem/14391>

- 어차피 3자리 수는 항상 4자리 수보다 작다
- 정답은 네자리 수 4개 합이 아닐까?

0	0	0	5
0	0	0	0
0	0	0	0
5	0	0	0

0	0	0	5
0	0	0	0
0	0	0	0
5	0	0	0

종이 조각

<https://www.acmicpc.net/problem/14391>

- 어차피 3자리 수는 항상 4자리 수보다 작다
- 정답은 네자리 수 4개 합이 아닐까? 아니다
- 반례가 있다

0	0	0	5
0	0	0	0
0	0	0	0
5	0	0	0

종이 조각

<https://www.acmicpc.net/problem/14391>

- $1 \leq N, M \leq 4$ 이다.
- 각각의 칸은 가로 또는 세로 칸에 속하게 된다

4	9	3	7
2	5	9	1
3	8	4	6
9	1	5	0

종이 조각

<https://www.acmicpc.net/problem/14391>

- 각각의 칸에 대해서, 가로(-)인지 세로(|)인지 정하면 된다.

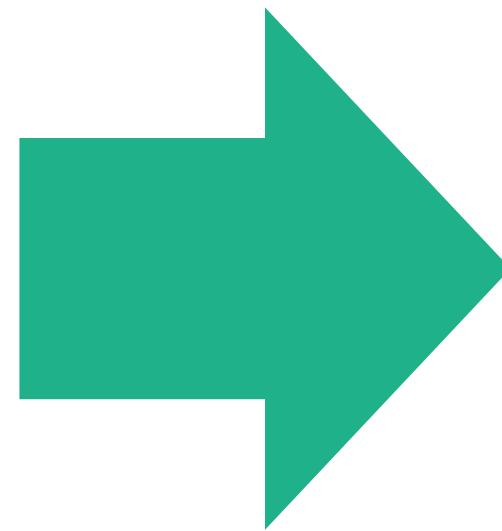
-	-		
-	-	-	
	-		
	-		-

종이 조각

<https://www.acmicpc.net/problem/14391>

- 각각의 칸에 대해서, 가로(-)인지 세로(|)인지 정하면 된다.

-	-		
-	-	-	
	-		
	-		-



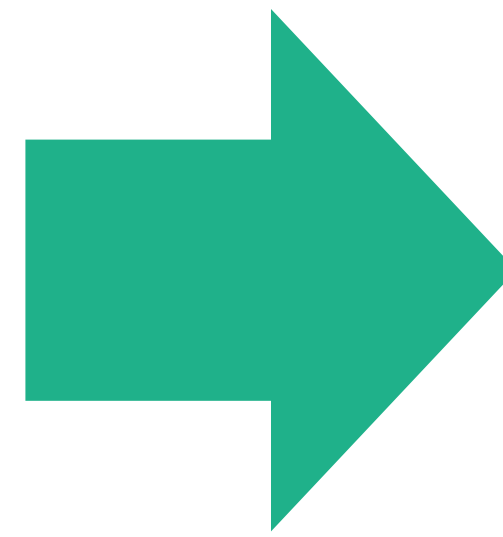
0	0	1	1
0	0	0	1
1	0	1	1
1	0	1	0

종이 조각

<https://www.acmicpc.net/problem/14391>

- 각각의 칸에 대해서, 가로(-)인지 세로(|)인지 정하면 된다.

-	-		
-	-	-	
	-		
	-		-



0	0	1	1
0	0	0	1
1	0	1	1
1	0	1	0

0	0	1	1	0	0	0	1	1	0	1	1	1	0	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

종이 조각

<https://www.acmicpc.net/problem/14391>

- 2^{NM} 으로 상태를 만들고 나누어보면 된다

종이 조각

<https://www.acmicpc.net/problem/14391>

- 소스: <http://codeplus.codes/0723f85d7e7c4a99a150ba83796c1432>

끝

코드 플러스

<https://code.plus>

- 슬라이드에 포함된 소스 코드를 보려면 "정보 수정 > 백준 온라인 저지 연동"을 통해 연동한 다음, "백준 온라인 저지"에 로그인해야 합니다.
- 강의 내용에 대한 질문은 코드 플러스의 "질문 게시판"에서 할 수 있습니다.
- 문제와 소스 코드는 슬라이드에 첨부된 링크를 통해서 볼 수 있으며, "백준 온라인 저지"에서 서비스됩니다.
- 슬라이드와 동영상 강의는 코드 플러스 사이트를 통해서만 볼 수 있으며, 동영상 강의의 녹화와 다운로드, 배포와 유통은 저작권법에 의해서 금지되어 있습니다.
- 다른 경로로 이 슬라이드나 동영상 강의를 본 경우에는 codeplus@startlink.io 로 이메일 보내주세요.
- 강의 내용, 동영상 강의, 슬라이드, 첨부되어 있는 소스 코드의 저작권은 스타트링크와 최백준에게 있습니다.