

브루트 포스 – 비트마스크 (연습)

최백준 choi@startlink.io

부분수열의 합

<https://www.acmicpc.net/problem/14225>

- 수열 S 가 주어졌을 때, 수열 S 의 부분 수열의 합으로 나올 수 없는 가장 작은 자연수를 구하는 문제
- 예를 들어, $S = [5, 1, 2]$ 인 경우에 1, 2, 3(=1+2), 5, 6(=1+5), 7(=2+5), 8(=1+2+5)을 만들 수 있다. 하지만, 4는 만들 수 없기 때문에 정답은 4이다.

부분수열의 합

<https://www.acmicpc.net/problem/14225>

- S의 부분 수열의 개수는 2^N 가지
- $N \leq 20$ 이기 때문에, 부분 수열을 모두 만들어 본다
- 부분 수열을 만드는 방법
 1. 재귀 호출
 2. 비트마스크

부분수열의 합

<https://www.acmicpc.net/problem/14225>

- 소스: <http://codeplus.codes/38a4d8336891434f8acac079388e7e52>

가르침

<https://www.acmicpc.net/problem/1062>

- N개의 단어가 주어졌을 때
- K개의 글자로만 이루어진 단어의 개수를 고르는 문제
- 모든 단어는 anta로 시작하고
- 모든 단어는 tica로 끝난다
- $N \leq 50$, 단어의 길이 ≤ 15

가르침

<https://www.acmicpc.net/problem/1062>

- N개의 단어가 주어졌을 때
- K개의 글자로만 이루어진 단어의 개수를 고르는 문제
- 모든 단어는 **anta**로 시작하고
- 모든 단어는 **tica**로 끝난다
- $N \leq 50$, 단어의 길이 ≤ 15
- 먼저, a, n, t, i, c는 가르쳐야 한다
- 즉, 26-5개의 글자 중에서 K-5개를 고르는 문제

가르침

<https://www.acmicpc.net/problem/1062>

- 경우의 수가 몇 개 되지 않기 때문에, 모든 경우를 다 시뮬레이션 해본다

가르침

<https://www.acmicpc.net/problem/1062>

```
int go(int index, int k, vector<string> &words) {
    if (k < 0) return 0;
    if (index == 26) return count(words);
    int ans = 0;
    learn[index] = true;
    int t1 = go(index+1, k-1, words);
    learn[index] = false;
    if (ans < t1) ans = t1;
    if (index != 'a'-'a' && index != 'n'-'a' && index != 't'-'a'
    && index != 'i'-'a' && index != 'c'-'a') {
        t1 = go(index+1, k, words);
        if (ans < t1) ans = t1;
    }
    return ans;
}
```


가르침

<https://www.acmicpc.net/problem/1062>

```
int count(vector<string> &words) {  
    int cnt = 0;  
    for (string word : words) {  
        bool ok = true;  
        for (char x : word) {  
            if (!learn[x-'a']) {  
                ok = false;  
                break;  
            }  
        }  
        if (ok) cnt += 1;  
    }  
    return cnt;  
}
```

가르침

10

<https://www.acmicpc.net/problem/1062>

- 소스: <http://codeplus.codes/a5f5133261494434ba34dfaec4a8ca2b>

가르침

<https://www.acmicpc.net/problem/1062>

- 각각의 단어가 배운 알파벳으로만 이루어져 있는지 검사하는 것이 너무 오랜 시간이 걸린다.
- $O(\text{단어의 개수} \times \text{각 단어의 길이})$
- 이 부분을 $O(\text{단어의 개수})$ 로 줄일 수 있다.

가르침

<https://www.acmicpc.net/problem/1062>

- 실제로 그 단어가 무엇인지가 중요한 것이 아니다.
- 그 단어에 알파벳이 어떤 순서로 이루어져 있는지 중요한 것이 아니다.
- 각 단어에 속해있는 알파벳이 무엇인지만 중요하다.

가르침

<https://www.acmicpc.net/problem/1062>

- 각 단어를 비트마스크로 나타낼 수 있다.
- $\text{word}[i]$ = i 번째 단어에 포함되어 있는 알파벳의 비트마스크

가르침

<https://www.acmicpc.net/problem/1062>

```
vector<int> words(n);  
for (int i=0; i<n; i++) {  
    string s;  
    cin >> s;  
    for (char x : s) {  
        words[i] |= (1 << (x-'a'));  
    }  
}
```

가르침

<https://www.acmicpc.net/problem/1062>

```
int go(int index, int k, int mask, vector<int> &words) {
    if (k < 0) return 0;
    if (index == 26) return count(mask, words);
    int ans = 0;
    int t1 = go(index+1, k-1, mask | (1 << index), words);
    if (ans < t1) ans = t1;
    if (index != 'a'-'a' && index != 'n'-'a' && index != 't'-'a'
    && index != 'i'-'a' && index != 'c'-'a') {
        t1 = go(index+1, k, mask, words);
        if (ans < t1) ans = t1;
    }
    return ans;
}
```

가르침

16

<https://www.acmicpc.net/problem/1062>

```
int count(int mask, vector<int> &words) {  
    int cnt = 0;  
    for (int word : words) {  
        if ((word & ((1<<26)-1-mask)) == 0) {  
            cnt += 1;  
        }  
    }  
    return cnt;  
}
```


가르침

<https://www.acmicpc.net/problem/1062>

- 소스: <http://codeplus.codes/2a8a54ed0b1044528417447a9694e4dc>

구슬 탈출 2

<https://www.acmicpc.net/problem/13460>

- 보드의 상태가 주어졌을 때, 최소 몇 번 만에 빨간 구슬을 구멍을 통해 빼낼 수 있는지 구하는 문제
- 만약, 10번 이내에 움직여서 빨간 구슬을 구멍을 통해 빼낼 수 없으면 -1을 출력

구슬 탈출 2

<https://www.acmicpc.net/problem/13460>

- 보드의 상태가 주어졌을 때, 최소 몇 번 만에 빨간 구슬을 구멍을 통해 빼낼 수 있는지 구하는 문제
- 만약, **10번 이내**에 움직여서 빨간 구슬을 구멍을 통해 빼낼 수 없으면 -1을 출력

구슬 탈출 2

20

<https://www.acmicpc.net/problem/13460>

- 이동할 수 있는 방향이 4방향
- 최대 10번 이내로 움직여야 한다
- 가능한 이동 방법의 수: $4^{10} = 1,048,576$

구슬 탈출 2

<https://www.acmicpc.net/problem/13460>

- 같은 방향으로 연속해서 두 번 이상 이동하는건 의미가 없다
- 가능한 이동 방법의 수: $4 \times 3^9 = 78,732$ 가지

구슬 탈출 2

<https://www.acmicpc.net/problem/13460>

- 같은 방향으로 연속해서 두 번 이상 이동하는건 의미가 없다
- 한 방향으로 이동한 다음, 반대 방향으로 바로 이동하는 것도 의미가 없다
- 가능한 이동 방법의 수: $4 \times 2^9 = 2,048$ 가지

구슬 탈출 2

<https://www.acmicpc.net/problem/13460>

- 먼저, 이동 가능한 방법을 비트마스크를 이용해서 4^{10} 가지를 만든 다음
- 앞 페이지에 나온 두 가지 경우를 모두 제외시킨다
- 4^{10} 을 만들기 위해 0부터 2^{20} 까지 수를 모두 만들고
- 4진법으로 변환해서 경우의 수를 모두 만든다

구슬 탈출 2

<https://www.acmicpc.net/problem/13460>

- 그 다음, 문제에 나와있는 대로 시뮬레이션 해본다.
- 동시에 두 개의 공을 이동시키는 것은 어렵기 때문에
- 공을 하나씩 움직여서 더 이상 두 공이 움직이지 않을 때까지 이동시켜본다

구슬 탈출 2

25

<https://www.acmicpc.net/problem/13460>

- 소스: <http://codeplus.codes/8881bdef1d394340ab85654ee36dac13>

2048 (Easy)

<https://www.acmicpc.net/problem/12100>

- 2048 게임에서 최대 5번 이동시켜서 얻을 수 있는 가장 큰 블록을 출력하는 문제

2048 (Easy)

<https://www.acmicpc.net/problem/12100>

- 이동 횟수가 5번이기 때문에, $4^5 = 1024$ 번 이동을 시켜보면 된다.

2048 (Easy)

<https://www.acmicpc.net/problem/12100>

28

- 소스: <http://codeplus.codes/196e4485f5214999a508fbfee034d63f>