

SIT/2021

Week 3

- 로또 -

안태진(taejin7824@gmail.com)

GitHub(github.com/taejin1221)

상명대학교 소프트웨어학과

201821002

Contents

- Approach
- Code

Approach

- 문제
 - k개 중에 6개를 고르는 모든 경우의 수를 구하라!
- 접근
 - 역시 일단은 Bruteforcing이지!
 - k가 최대 13이네
 - 그럼 $C(13, 6)$ (Combination)이고 1716이니 씹가능이네
 - ㄱㄱ

Approach

- 점화식

- $BT(start, end)$

- $start$: 시작점

- end : 끝점

- $BT(start, end) = \begin{cases} \emptyset & (if\ start == end) \\ \{start, BT(start + 1, end)\} \end{cases}$

Code

- 헤더파일 선언
 - list가 필요하기 때문에 vector 사용
- 전역 변수 선언
 - Backtracking 해야하기 때문
 - n : 숫자의 갯수
 - arr: 숫자들
 - ans: 수를 고르는 방법들

```
#include <iostream>
#include <vector>
using namespace std;
```

```
int n;
int arr[13];
vector<vector<int>> ans;
```

Code

- Backtracking 함수
 - temp: 이전에 만든 조합
 - idx: 시작 index
- 6개 다 뽑았다면 ans에 조합 추가
- start부터 끝까지 dfs 시작

```
void dfs( vector<int>& temp, int idx ) {
```

```
    if ( temp.size() == 6 )  
        ans.push_back( temp );
```

```
    else {  
        for ( int i = idx; i < n; i++ ) {  
            temp.push_back( arr[i] );  
            dfs( temp, i + 1 );  
            temp.pop_back( );  
        }  
    }
```

Code

- 빠른 입출력
 - 입력이 여러 테스트고 13개 입력
 - 출력도 엄청 많이 출력
- 입력과 테스트 케이스 시작
- 정답 list 초기화 및 Backtracking 시작
 - 모든 테스트 케이스에서 같은 list를 사용

```
int main(void) {  
    ios_base::sync_with_stdio(false);  
    cin.tie(NULL); cout.tie(NULL);
```

```
    cin >> n;  
    while ( n != 0 ) {  
        for ( int i = 0; i < n; i++ )  
            cin >> arr[i];
```

```
        ans.clear();  
        vector<int> temp;  
        dfs( temp, 0 );
```

Code

- 출력
 - 작은 것부터 큰 순으로 추가하기 때문에 자동으로 사전순으로 정렬되어있음
- 다음 테스트 케이스를 위해 입력

```
for ( int i = 0; i < ans.size(); i++ ) {  
    for ( int& value : ans[i] )  
        cout << value << ' ';  
    cout << '\n';  
}  
cout << '\n';
```

```
cin >> n;
```

감사합니다!
