



# 09. 완전탐색 & 재귀호출

Div. 3 알고리즘 스터디 / 임지환



# 완전탐색?

- 가능한 모든 경우의 수를 일일이 나열하면서 답을 찾는 방법
- 주어지는 입력의 크기에 따라 가능여부가 다름



# 완전탐색의 예시

1. 1~N까지의 합
2. N명의 사람을 줄세우는 방법의 수
3. 서로 다른 N개의 숫자로 N자리 자연수를 만들 때 어떤 N자리 수 k가 사전순으로 나열했을 때 몇번째에 위치하는가?
4. 1~N까지의 도시가 있고 각 도시간 이동 비용이 주어질 때, 1번도시부터 출발하여 모든 도시를 거쳐 다시 1번 도시로 돌아오는 경로 중 가장 적은 비용



ex 1) 1~N까지의 합



## ex 1) 1~N까지의 합

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



## ex 1) 1~N까지의 합

```
int sum2(int n) {  
    return (n * (n + 1)) / 2;  
}
```



## ex 1) 1~N까지의 합

- dp때 했던 것처럼 부분 문제로 쪼갠다면?



## ex 1) 1~N까지의 합

- dp때 했던 것처럼 부분 문제로 쪼갠다면?  
 $\Rightarrow N + 1 \sim N-1$ 까지의 합





## ex 1) 1~N까지의 합

- dp때 했던 것처럼 부분 문제로 쪼갰다면?

⇒  $N + 1 \sim N-1$ 까지의 합

```
int sum3(int n) {  
    if (n == 1) return 1;  
    return n + sum3(n - 1);  
}
```



# #2309 일곱 난쟁이

## 문제

---

왕비를 피해 일곱 난쟁이들과 함께 평화롭게 생활하고 있던 백설공주에게 위기가 찾아왔다. 일과를 마치고 돌아온 난쟁이가 일곱 명이 아닌 아홉 명이었던 것이다.

아홉 명의 난쟁이는 모두 자신이 "백설 공주와 일곱 난쟁이"의 주인공이라고 주장했다. 뛰어난 수학적 직관력을 가지고 있던 백설공주는, 다행스럽게도 일곱 난쟁이의 키의 합이 100이 됨을 기억해 냈다.

아홉 난쟁이의 키가 주어졌을 때, 백설공주를 도와 일곱 난쟁이를 찾는 프로그램을 작성하시오.

## 입력

---

아홉 개의 줄에 걸쳐 난쟁이들의 키가 주어진다. 주어지는 키는 100을 넘지 않는 자연수이며, 아홉 난쟁이의 키는 모두 다르며, 가능한 정답이 여러 가지인 경우에는 아무거나 출력한다.



# #2309 일곱 난쟁이

- 다행스럽게도 일곱 난쟁이의 키의 합이 100이 된다.



## #2309 일곱 난쟁이

- 다행스럽게도 일곱 난쟁이의 키의 합이 100이 된다.
- 더 다행스럽게도 돌아온 난쟁이의 수는 9명이다. 7보다 큰 임의의 자연수가 아니다.



## #2309 일곱 난쟁이

- 다행스럽게도 일곱 난쟁이의 키의 합이 100이 된다.
- 더 다행스럽게도 돌아온 난쟁이의 수는 9명이다. 7보다 큰 임의의 자연수가 아니다.
- 9명 중 2명만 걸러내면 원하는 7명이 나올 것이다.



# 스포 방지용 슬라이드






## #2309 일곱 난쟁이

- 1) 9명의 키의 합을 구해놓는다.
- 2)  $1 \leq i, j \leq 9$ 인 서로 다른  $i, j$ 의 순서쌍을 모두 구해보며 합에서 그 값을  $i$ 의 키와  $j$ 의 키를 빼본다.
- 3) 그 차이가 100이 된다면 탐색을 중단하고  $i$ 번째와  $j$ 번째 난쟁이의 키가 아닌 것을 오름차순으로 출력한다.



## #2309 일곱 난쟁이

- 1) 9명의 키의 합을 구해놓는다.
- 2)  $1 \leq i, j \leq 9$  인 서로 다른  $i, j$ 의 순서쌍을 모두 구해보며 합에서 그 값을  $i$ 의 키와  $j$ 의 키를 빼본다.  2중 for문으로 충분!
- 3) 그 차이가 100이 된다면 탐색을 중단하고  $i$ 번째와  $j$ 번째 난쟁이의 키가 아닌 것을 오름차순으로 출력한다.

모든 경우의 수 :  ${}_9C_2$





# #15650 N과 M (2)

## 문제

---

자연수 N과 M이 주어졌을 때, 아래 조건을 만족하는 길이가 M인 수열을 모두 구하는 프로그램을 작성하시오.

- 1부터 N까지 자연수 중에서 중복 없이 M개를 고른 수열
- 고른 수열은 오름차순이어야 한다.

## 입력

---

첫째 줄에 자연수 N과 M이 주어진다. ( $1 \leq M \leq N \leq 8$ )



## #15650 N과 M (2)

- 앞의 문제와 다르게 M값이 확실치가 않다.
  - M중 for문을 사용할 수 없다ㅠㅠ
- "오름차순"이라는 규칙이 있다.



# 스포 방지용 슬라이드





## #15650 N과 M (2)

- 1) N과 M이 어떻게 주어져도 경우의 수는  $N C_M$ 이다.
- 2) 전체 문제와 부분 문제를 정의한다.
  - 전체 문제 : 1~N에서 서로 다른 M개를 뽑는 경우의 수
  - 부분 문제 : 1~N에서 하나를 제외한 수들 중 서로 다른 M-1개를 뽑는 경우의 수
- 3) 오름차순은? 상관없다.



## #15650 N과 M (2)

```
void func(int cnt, int chosen[], int last) {
    if (cnt == m) {
        for (int i = 1; i <= n; i++)
            if (chosen[i]) printf("%d ", i);
        printf("\n");
        return;
    }
    for (int i = last + 1; i <= n; i++) {
        chosen[i] = 1;
        func(cnt + 1, chosen, i);
        chosen[i] = 0;
    }
}
```



# #1182 부분 수열의 합

## 문제

---

$N$ 개의 정수로 이루어진 수열이 있을 때, 길이가 양수인 부분수열 중에서 그 수열의 원소를 다 더한 값이  $S$ 가 되는 경우의 수를 구하는 프로그램을 작성하시오.

## 입력

---

첫째 줄에 정수의 개수를 나타내는  $N$ 과 정수  $S$ 가 주어진다. ( $1 \leq N \leq 20$ ,  $|S| \leq 1,000,000$ ) 둘째 줄에  $N$ 개의 정수가 빈 칸을 사이에 두고 주어진다. 주어지는 정수의 절댓값은 100,000을 넘지 않는다.



# #1182 부분 수열의 합

- 원소의 개수는 최대 20개
- 20개의 원소로 만들 수 있는 합의 종류는 최악의 경우  $2^{20}$ 개



# 스포 방지용 슬라이드







# #1182 부분 수열의 합

- 1) 각 원소들에 대해 선택할지, 안할지에 대한 선택을 내릴 수 있다.
- 2) 0~ N-1까지의 모든 index를 전부 고려한 후 sum을 봤을 때 우리가 구하고자 하는 수라면 count!

```
int n, s, ans, arr[20];
void work(int i, int sum) {
    sum += arr[i];
    if (i >= n) return;
    if (s == sum) ans++;
    work(i + 1, sum - arr[i]);
    work(i + 1, sum);
}
```



# 가만 보니 DFS랑 유사한데요?

맞습니다.



# Problem set

- 2309 일곱 난쟁이
- 1759 암호 만들기
- 13908 비밀번호
- 15649 N과 M (1)
- 15650 N과 M (2)
- 1182 부분 수열의 합
- 14502 연구소(삼성 기출)
- 1038 감소하는 수\*
- 1208 부분 수열의 합 2\*
- 10971 외판원 순회 2\*