KCPC 2022

Official Solutions

2023년 2월 4일

KCPC mini 2022

운영

- ✓ 김준겸 `ryute` 고려대학교 컴퓨터학과
- ✓ 노윤헌 `edwardblue` 고려대학교 컴퓨터학과

KCPC mini 2022

출제

- ✓ 김동현 kdh9949 — 서울대학교 컴퓨터공학부
- ✓ 김준겸 ryute — 고려대학교 컴퓨터학과
- ✓ 나정휘 jhnah917 — 숭실대학교 컴퓨터학부
- ✓ 노윤헌 edwardblue — 고려대학교 컴퓨터학과
- ✓ 오주원 kyo20111 — 숭실대학교 소프트웨어학부

KCPC mini 2022

검수

✓ 박찬솔	chansol	숭실대학교 컴퓨터학부
✓ 윤시우	cgiosy	서울사이버대학교 컴퓨터공학과
✓ 이종영	moonrabbit2	KAIST 전산학부
✓ 정희승	gs18103	서울대학교 물리천문학부
✓ 최은수	gs18115	KAIST 전산학부

Sponsors

KCPC mini 2022

KCPC mini 2022

문제		의도한 난이도	출제자
2A	금공강 사수	2	edwardblue
2B	참살이길	5	edwardblue
2C	읽씹 멈춰!	5	edwardblue
2D/1D	아 또 XOR이야?	1	ryute
2E/1F	산유국	4	kyo20111
2F/1A	단조증가 수열과 OR	2	kdh9949
1B	순찰 경로	1	jhnah917
1E	화살표 수집가	2	kdh9949
1C	알록달록 트리	3	edwardblue

KCPC mini 2022

수상자

- Div 1 최우수상 (노세윤) 상품 : 에어팟 프로 2
- Div 1 우수상 (강근호) 상품 : 15만원
- Div 1 장려상 (이재호) 상품 : 7만원
- Div 2 최우수상 (김재우) 상품 : 에어팟 프로 2
- Div 2 우수상 (박준현, 조영민) 상품 : 키보드 (Deck CBL-87XN 거북선, 적축)
- Div 2 장려상 (최준혁, 조영준, 한규식, 윤성하) 상품 : 마우스
- 협찬 문제 Div. 2B-참살 이길 First Solve(최기성) 상품 : 마우스
- Div. 1 가장 많은 제출로 정답을 받은 참가자 (강인구) 상품 : 마우스
- Div. 1, 2 가장 마지막 정답자 (김상엽) 상품 : 마우스

KCPC mini 2022

2A. 금공강 사수

`bruteforcing`

출제진 의도 – 

- ✓ 처음 푼 사람: **이창엽**, 6분
- ✓ 처음 푼 사람(Open): **amel**, 2분
- ✓ 출제자: `edwardblue`

A. 금공강 사수

KCPC mini 2022

- n이 20이하이므로 최대 2^{20}가지의 경우만 탐색하면 됩니다.
- 따라서 완전 탐색(브루트 포스)로 해결 가능합니다.
- 시간표를 나타내는 2차원 배열 table[5][10]을 만듭니다.
- table[i][j]는 i번째 요일의 j교시에 수업이 있는지를 나타냅니다.
- 한 수업은 최대 10교시이므로 어떤 수업이 들어왔을 때 table은 최대 10칸 수정됩니다.
- 따라서 전체 시간 복잡도는 $\mathcal{O}(10 \times 2^n)$입니다.

KCPC mini 2022

2B. 참살이길

`sorting`, `math`

출제진 의도 – 

- 처음 푼 사람: **최기성**, 40분
- 처음 푼 사람(Open): **amel**, 9분
- 출제자: `edwardblue`

KCPC mini 2022

- 이 문제는 신호등을 x좌표 순서대로 정렬한 뒤, 참살이길의 전체 길이에 각 신호등에서 얼마나 기다려야 하는지를 더해 주면 되는 문제입니다.
- i번째 신호등에서 기다려야 하는 시간은, i번째 신호등에 도달한 순간부터 그 신호등이 초록 불이 될 때까지 걸리는 시간입니다. s_i초 직후 i번째 신호등은 초록 불이 되므로, 이후 i번째 신호등이 초록 불이 되는 시간은 $s_i, s_i + 2t_i, s_i + 4t_i, \ldots$입니다.
- i번째 신호등까지 도달하는 데 걸린 시간을 T라고 하면, 이 신호등이 초록 불이 되고 나서부터 지난 시간은 $p_i = (T - s_i) \mod 2t_i$입니다. 따라서, p_i의 값이 0 이상 t_i 미만이라면 초록 불이므로 그냥 지나가면 되고, t_i 이상 $2t_i$ 미만이라면 $2t_i - p_i$초를 기다려야 하므로 T에 $2t_i - p_i$를 더해 줍니다.

KCPC mini 2022

2C. 읽씹 멈춰!

`greedy`, `math`

출제진 의도 – 

- 처음 푼 사람: **김재우**, 25분
- 처음 푼 사람(Open): **hsh8086**, 4분
- 출제자: `edwardblue`

KCPC mini 2022

- 문제를 단순화시키면, 0에서 시작하여 1을 더하는 연산과 2를 곱하는 연산을 반복하여 원하는 숫자 n을 가장 적은 비용으로 만드는 문제라고 볼 수 있습니다.
- 편의를 위해 $+1$ 연산, $\times 2$ 연산이라고 합시다.
- 우선 최소 비용으로 n을 만드는 정답 수열이 있다고 가정합시다.
- 이 정답 수열에 $\times 2$가 존재한다면, 수열에서 가장 나중에 나오는 $\times 2$를 생각할 수 있습니다.
- 마지막 $\times 2$ 이후에 $+1$ 연산이 연속 2번 존재한다고 가정합시다.
- 이는 정답 수열이 될 수 없습니다.

KCPC mini 2022

- 그 이유는 마지막 $\times 2$ 직전에 $+1$ 연산을 하고, $\times 2$를 하면 $+1$ 연산을 1번만 쓰고도 같은 값을 낼 수 있기 때문입니다.
- 즉, 마지막 $\times 2$ 이후에 $+1$ 연산은 최대 1개입니다.
- 같은 방식으로 생각해 보면, $\times 2$ 연산과 $\times 2$ 연산 사이에는 $+1$ 연산이 최대 1개입니다.
- 2개 이상이라면 앞의 $\times 2$ 연산 직전에 $+1$ 연산을 하는 것이 더 이득이기 때문입니다.

KCPC mini 2022

- 이제 해답을 내기 위해 n에서 $\div 2$, -1을 반복해 0에 도달한다고 생각해 봅시다.
- 전 슬라이드의 증명에 의해, $\div 2$와 $\div 2$ 사이의 -1은 최대 1개입니다.
- n에서 $\div 2$, -1를 반복하다가 홀수에 도달했다면, 반드시 -1을 사용해야만 합니다.
- n에서 $\div 2$, -1를 반복하다가 짝수에 도달했다면, 두 가지 경우를 생각해 볼 수 있습니다.
- -1 연산을 선택하는 경우, 앞으로는 $\div 2$를 더 이상 사용할 수 없습니다.
- 연산을 적용한 결과 남은 숫자가 홀수가 되므로, -1 연산을 한 번 더 적용해야 하기 때문입니다.
- 결국 -1을 두 번 연속 사용하게 되므로 $\div 2$를 더 이상 사용할 수 없습니다.
- $\div 2$ 연산을 선택하는 경우에는 다시 위의 두 가지 경우를 생각하면 됩니다.

KCPC mini 2022

- 결국 우리는 어떤 짝수 n에 도달했을 때, $\div 2$를 사용할 지 -1을 사용할 지만 정하면 됩니다.
- -1을 사용하는 경우 -1을 n번 해야만 0에 도달할 수 있습니다.
- 이 경우 드는 비용은 ns입니다.
- $\div 2$를 사용하는 경우 드는 비용은 t입니다.
- 만약 $\frac{ns}{2} \leq t$라면 더 이상 $\div 2$를 사용하는 것은 손해입니다.
- 만약 $\frac{ns}{2} > t$라면, $\div 2$를 먼저 사용하는 것이 항상 이득입니다.

- ✓ 결국 다음과 같은 greedy 해법을 생각할 수 있습니다.
- ✓ n에서 출발하여 홀수면 -1, 짝수면 $\frac{ns}{2} > t$ 일때까지는 $\div 2$를 하고 $\frac{ns}{2} \leq t$ 이면 남은 숫자는 -1만을 사용하여 0으로 만든다.
- ✓ 시간복잡도는 $\mathcal{O}(T \log n)$ 입니다.

KCPC mini 2022

2D/1D. 아 또 XOR이야?

`math, dynamic programming`

출제진 의도 – 

- ✓ 처음 푼 사람: **???**, ?분 / **박홍빈**, 51분
- ✓ 처음 푼 사람(Open): **hyperbolic**, 18분
- ✓ 출제자: `Ryute`

KCPC mini 2022

- ✓ A 이상 B 이하에서 개수를 세는 건 B 이하의 개수에서 $A-1$ 이하의 개수를 빼는 것과 같습니다.
- ✓ 따라서 A 이하이면서 조건을 만족하는 수를 세는 문제로 바꿉시다.

- ✓ A 이하인 수들을 두 가지로 분류해봅시다.
- ✓ 만약 A의 가장 큰 비트 위치와 같은 위치에 0이 있으면, 이 수는 반드시 A보다 작습니다.
- ✓ 만약 A의 가장 큰 비트 위치와 같은 위치에 1이 있으면, 가장 큰 비트를 무시하고 나머지 비트에 대해서만 비교해도 결과가 같습니다.

KCPC mini 2022

- 비트가 0인 경우를 고려해봅시다.
- 일단 N과 가장 큰 비트가 차이 나니 K가 1만큼 줄어듭니다.
- 이 수들은 모두 A보다 작으니, 차이가 정확히 비트 K개 만큼 나는 수의 개수만 세면 됩니다.
- 뒤의 비트가 무엇이던 간에 N과 서로 다른 비트를 K개 고르는 것과 같으므로 남은 비트의 수가 T개 일 때 $\binom{T}{K}$를 계산해 더해주면 됩니다.

KCPC mini 2022

- ✓ 비트가 1 인 경우를 고려해봅시다.
- ✓ 이 경우는 가장 큰 비트를 떼도 문제가 바뀌지 않으므로 재귀적으로 계산해주면 됩니다.

KCPC mini 2022

2E/1F. 산유국

`sweeping`, `prefix_sum`, `tree_set`, `deque`

출제진 의도 – 

- 처음 푼 사람: **김재우**, 170분 / **박홍빈**, 104분
- 처음 푼 사람(Open): **hyperbolic**, 48분
- 출제자: kyo20111

KCPC mini 2022

- 위협 단체가 존재하는 구역의 개수를 M 이라고 합시다.
- K 개 이상의 위협 단체가 포함된 구역을 점령하는 게 아니라 $M - K$ 개 이하의 위협 단체가 존재하는 구간을 제외한 나머지 구역들을 점령하는 것으로 생각할 수 있습니다.
- 즉, (모든 구역의 합) $-$ ($M - K$ 개 이하의 위협 단체가 존재하는 구간의 합 중 최소)가 답이라고 생각할 수 있습니다.

KCPC mini 2022

- ✓ A와 B에 각각의 복사본을 뒤에 붙여 원형이 아닌 길이 $2N$의 나라라고 생각해봅시다.
  - $A_1 = A_{N+1}, A_2 = A_{N+2}, \ldots, A_N = A_{N+N}$
  - $B_1 = B_{N+1}, B_2 = B_{N+2}, \ldots, B_N = B_{N+N}$
- ✓ $N+1$부터 $N+N$까지 각각의 구역이 구간의 가장 오른쪽일 때, $M-K$개 이하의 위협 단체가 존재하는 구간이도록 선택할 수 있는 가장 왼쪽 구역인 L_i을 찾습니다.
  - B_i가 1인 구역이 K개 이상이라는 조건이 있으므로 $i - L_i + 1 < N$을 만족합니다.
  - $L_i \le L_{i+1}$이기 때문에 $N+1$부터 $N+N$까지 순서대로 본다면 선형 시간에 L_i를 모두 구할 수 있습니다.

KCPC mini 2022

- ✓ A의 누적합 배열 S를 만들었다고 합시다.
  - $S_i = S_i + A_{i-1}$
- ✓ i번이 가장 오른쪽인 구간이 가질 수 있는 합의 최솟값은 $S_i - \max(S_0, S_1, \ldots, S_{i-1})$ 입니다.
- ✓ $M - K$ 이하의 위협 단체를 포함해야 한다는 조건을 만족하면서 i번이 가장 오른쪽인 구간이 가질 수 있는 합의 최솟값은 $S_i - \max(S_{L_i-1}, S_{L_i}, \ldots, S_{i-1})$ 입니다.

KCPC mini 2022

- 각 i 마다 $S_{L_i-1}, S_{L_i}, \ldots, S_{i-1}$ 를 저장하면서 그 중 최댓값을 가져오는 연산을 할 수 있으면 됩니다.
- 이는 L_i 를 선형 시간에 구하듯이 S_{i-1} 를 삽입하고, $S_{L_{i-1}-1}, S_{L_{i-1}}, \ldots, S_{L_i-2}$ 를 삭제하는 연산을 하면 원하는 구간의 값만 저장할 수 있습니다.
- 이때, S_i 는 최대 한번 삽입되고 최대 한번 삭제됩니다.
- 그러므로 multiset을 이용한다면 삽입과 삭제를 $O(N)$ 번 $\mathcal{O}(\log N)$ 에 하고, 최댓값을 $O(N)$ 번 $\mathcal{O}(1)$ 에 확인할 수 있으므로 시간복잡도 $\mathcal{O}(N \log N)$ 에 문제를 해결할 수 있습니다.

KCPC mini 2022

- 먼저 삽입된 값이 먼저 삭제된다는 성질을 이용해 deque로 $\mathcal{O}(N)$에 문제를 해결할 수 있습니다.
- 구간 $[L, R]$이 주어졌을 때, $L \leq i \leq j \leq R$의 구간 $[i, j]$ 중 최솟값을 구하는 쿼리를 세그먼트 트리를 이용해 $\mathcal{O}(\log N)$에 구할 수 있습니다.
- 이를 이용해 L_i를 구하고 $[L_i, i]$ 쿼리로 최솟값을 구해주는 방법으로 이 문제를 해결할 수 있습니다.

KCPC mini 2022

2F/1A. 단조증가 수열과 OR

greedy

출제진 의도 – 

- 처음 푼 사람: **???**, ?분 / **이재호**, 71분
- 처음 푼 사람(Open): **hyperbolic**, 83분
- 출제자: kdh9949

KCPC mini 2022

- ✓ 상위 비트부터 값을 결정해 줍시다.
- ✓ 이제 수열의 각 원소를 0 또는 1이라고 가정할 수 있습니다.
- ✓ 0과 1을 그리디하게 배치할 수 있는 방법이 있을까요?

KCPC mini 2022

- 단조증가 조건은 일단 고려하지 않고, 채워야 할 값을 먼저 채워 줍시다.
- $A[k][i]$, $B[k][i]$를 각각 A_i, B_i의 k번째 비트라고 합시다.
- 우선, $A[k][i] = 0$일 경우 $B[k][i]$, $B[k][i+1]$은 모두 0입니다.
- 이제, $A[k][i] = 1$인데 $B[k][i]$, $B[k][i+1]$ 둘 중 하나가 0인 경우 나머지 하나는 1입니다.
  - $B[k][i] = B[k][i+1] = 0$일 수 있는데, 이 때는 불가능한 것입니다.
- 이렇게 값을 채워 주고 나면, $B[k][i]$가 아직 결정되지 않은 경우는 $A[k][i-1]$, $A[k][i]$가 모두 1인 경우 뿐입니다.

KCPC mini 2022

- ✓ 이제 단조증가 조건을 고려하여 결정되지 않은 $B[k][i]$ 값들을 채워 봅시다.
- ✓ 이미 결정한 상위 비트들의 값에 따라, 현재 수열을 "단조증가해야 하는 구간"들로 나눌 수 있습니다.
- ✓ 각 구간 내에서 $B[k][i]$ 값들은 무조건 "$0, \ldots, 0, 1, \ldots, 1$" 과 같은 꼴이어야 합니다.
- ✓ 즉, 0이 1 뒤에 나타나면 안 됩니다. 불가능한 경우를 여기서 또 걸러 줍시다.
- ✓ 구간 내에서 0을 하나 발견하면, 그 앞의 수 역시 모두 0입니다.
- ✓ 마찬가지로, 1을 하나 발견하면 그 뒤에는 모두 1입니다.

KCPC mini 2022

- ✓ 여기까지 수행하고도 아직 결정되지 않은 $B[k][i]$ 값들이 존재할 수 있습니다.
- ✓ 그런 값들을 ?라고 하고, 구간 내에 있는 연속한 ?들의 덩어리에 대해 생각해 봅시다.
- ✓ 값을 채우는 과정을 잘 생각해 보면, 구간의 시작점을 포함하는 ? 덩어리만이 남아 있을 가능성이 있습니다.
  - (맨 왼쪽에 있지 않은) ? 덩어리 바로 왼쪽에 1이 반드시 존재하기 때문입니다.

KCPC mini 2022

- 이제 이 ? 덩어리를 채울 수 있는 꼴은 "$0, 1, \ldots$"과 "$1, 1, \ldots$"의 두 가지 뿐입니다. 그런데, 둘 중에 앞에 0을 넣는 쪽이 "단조증가해야 하는 구간"을 더 잘게 쪼개기 때문에 무조건 이득입니다.
- 구간의 길이가 1일 경우에는 0 말고 1을 채워야 함에 유의합시다.
- 위 전략대로 모든 ? 덩어리를 채우면 OR 조건을 무조건 만족시킬 수 있습니다.
- 이제 한 비트 아래로 내려가서 같은 일을 반복해주면 됩니다.
- 맨 아래 비트까지 다 채우는 데 성공했다면 답을 출력합니다.
- 시간복잡도는 $\mathcal{O}\left(N \log(\max A_i)\right)$ 입니다.

KCPC mini 2022

1B. 순찰 경로

`graphs`, `ad_hoc`, `constructive`

출제진 의도 – 

- ✓ 처음 푼 사람: **노세윤**, 43분
- ✓ 처음 푼 사람(Open): **amel**, 97분
- ✓ 출제자: `jhnah917`

KCPC mini 2022

- ✓ 주어진 트리가 성게 그래프이면 불가능합니다.
- ✓ 그렇지 않은 경우 항상 가능합니다.
- ✓ 다양한 풀이가 존재하는데, 두 가지 풀이를 소개합니다.

1B. 순찰 경로 - 풀이 1

KCPC mini 2022

- 시작 정점 s를 고정합시다.
- 트리는 s를 기준으로 몇 개의 서브 트리로 나누어 집니다.
- 매번 이전에 방문한 서브 트리와 다른 서브 트리로 이동하면 올바른 경로를 만들 수 있습니다.
- 하지만 어떤 서브 트리가 너무 크면 매번 다른 서브 트리로 이동하지 못할 수도 있습니다.
- 시작 정점에 따라 경로를 찾지 못할 수도 있기 때문에 잘 선택해야 합니다.

1B. 순찰 경로 - 풀이 1

KCPC mini 2022

- 만약 모든 서브 트리의 크기가 $N/2$ 이하라면 항상 올바른 경로를 찾을 수 있습니다.
- 모든 서브 트리의 크기를 $N/2$ 이하로 만드는 정점은 **센트로이드**라는 이름으로 잘 알려져 있습니다.
- 센트로이드에서 시작해서 매번 서로 다른 서브 트리로 이동하면 문제를 해결할 수 있습니다.

1B. 순찰 경로 - 풀이 2

KCPC mini 2022

- 트리는 이분 그래프입니다.
- 트리에서 깊이가 홀수인 정점과 짝수인 정점은 각자 서로 연결되어 있지 않습니다.
- 따라서 깊이가 홀수인 정점을 모두 방문한 다음 짝수 정점을 방문하면 됩니다.
- 깊이가 홀수인 정점에서 짝수인 정점으로 이동하는 부분을 잘 처리해야 합니다.
- 깊이가 $2, 4, \cdots, 2k, 1, 3, \cdots$ 인 정점들을 차례대로 방문하면 됩니다.

KCPC mini 2022

1E. 화살표 수집가

`geometry`, `offline_queries`

출제진 의도 – 2

- ✓ 처음 푼 사람: **???**, ?분
- ✓ 처음 푼 사람(Open): **amel**, 172분
- ✓ 출제자: kdh9949

KCPC mini 2022

- 점을 하나 고정해 봅시다.
- A를 고정하는 것이 가장 좋아 보입니다.
- 고정한 A에 대해 B, C 또는 D가 될 수 있는 점들만 남겨 봅시다.
- A를 기준으로 시계 12시 $\sim$ 9시 범위에 있는 점들만 남게 됩니다.

KCPC mini 2022

- ✓ 이 점들을 A 기준 반시계 방향으로 (9시 $\Rightarrow$ 12시) 정렬해 순서를 매깁시다. $P_1, P_2, \ldots, P_M$ 이라 하겠습니다.
- ✓ $B = P_i$ 로 정한다고 하면, C 와 D 로 가능한 P_j 들의 범위를 독립적으로 나타낼 수 있습니다.
- ✓ L_i 를 $j \leq i, \angle P_j A P_i < 90^\circ$ 인 최소 j 로 정의합시다.
- ✓ R_i 는 $j \geq i, \angle P_j A P_i < 90^\circ$ 인 최대 j 입니다.
- ✓ $L_i \leq L_{i+1}, R_i \leq R_{i+1}$ 이 성립하기 때문에, Two pointer 기법을 이용해 모든 L_i 및 R_i 값을 $\mathcal{O}(N)$ 에 구할 수 있습니다.
  - 세 점이 이루는 각도가 예각인지 판별하는 것은 벡터의 내적을 활용하면 됩니다.

KCPC mini 2022

- ✓ $D_i =$ (선분 AP_i의 길이) 라고 합시다.
- ✓ $B = P_i$일 때 만들 수 있는 화살표의 개수는 아래 두 값을 곱한 것과 같습니다.
  - $L_i \le j < i, D_j < D_i$인 j의 개수
  - $i < k \le R_i, D_k < D_i$인 k의 개수
- ✓ 점들을 D_i 순서대로 정렬하면 Offline query를 통해 $\mathcal{O}(N \log N)$에 답을 구할 수 있습니다.
- ✓ D_i 값이 같은 점들의 처리에 유의해야 합니다.
- ✓ 총 시간복잡도는 (A가 총 N개 있으니) $\mathcal{O}(N^2 \log N)$ 입니다.

KCPC mini 2022

1C. 알록달록 트리

`math`, `fft`

출제진 의도 – 

- ✓ 처음 푼 사람: **???**, ?분
- ✓ 처음 푼 사람(Open): **hyperbolic**, 202분
- ✓ 출제자: `edwardblue`

KCPC mini 2022

- 매우 쉬운 관찰을 하나 하고 가겠습니다.
- 트리의 각 정점을 색칠하는 경우의 수는 독립입니다.
- 따라서 임의의 정점을 색칠하는 경우의 수를 구한 뒤 모두 곱해 주면 답을 구할 수 있습니다.

KCPC mini 2022

- 이제 i번 정점이 d개의 자식을 갖고, 이 자식들을 l 이상 r 이하 가짓수의 색으로 칠하는 경우의 수를 구해 봅시다.
- 이 자식들이 정확히 c가지의 색으로 칠해져 있다면, i번 정점은 c개의 색 중 1개를 골라서 칠할 수 있습니다.
- 자식들을 정확히 c가지의 색으로 칠하는 가짓수는 어떻게 알 수 있을까요?
- 만약 d개의 서로 다른 정점을 c개의 서로 다른 색으로 색칠하되, 각 색깔은 한 번 이상씩 사용해야 하는 경우의 수를 구할 수 있다면, 우리는 여기에 k개의 색 중 c개의 색을 골라 주는 가짓수인 $\binom{k}{c}$를 곱해서 전체 가짓수를 알 수 있습니다.

KCPC mini 2022

- 제2종 스털링 수 $S(n, k)$는 서로 다른 n개의 원소를 k개의 구분되지 않는 부분집합으로 나누되, 각 부분집합은 한 개 이상의 원소를 갖도록 하는 경우의 수입니다.
- 여기서 각 부분집합이 서로 구분 가능한 경우에는 부분집합에 번호를 매겨 준다고 생각하면 되므로, $S(n, k) \cdot k!$가 전체 경우의 수가 됩니다.
- 자식들을 원소로 보고, 부분집합을 색으로 본다면 우리가 전 슬라이드에서 구하고 싶었던 가짓수와 정확히 일치합니다.

KCPC mini 2022

- 즉 d개의 자식을 갖는 i번 정점의 자식들을 정확히 c개의 색으로 칠하는 경우의 수는 $\binom{k}{c} \cdot S(d, c) \cdot c!$이 됩니다.
- 이제 i번 정점은 자식의 c개의 색 중 아무거나 사용할 수 있으므로 c를 곱해 주면, 자식들이 정확히 c개의 색으로 칠해져 있을 때 i번 정점을 색칠하는 경우의 수를 구할 수 있습니다.
- 결국 답은

$$\sum_{c=l}^{r} c \cdot \binom{k}{c} \cdot S(d, c) \cdot c!$$

입니다.

KCPC mini 2022

- ✓ 팩토리얼, 이항계수 등은 빠르게 구할 수 있으므로, 결국 제2종 스털링 수를 빠르게 구하면 문제를 해결할 수 있습니다.
- ✓ 제 2종 스털링 수는 다음과 같은 식을 만족합니다.

$$S(n,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^{k-i}\binom{k}{i}i^n$$

- ✓ 따라서 $S(n,0), ..., S(n,n)$을 한 번에 고속 푸리에 변환(FFT)로 $\mathcal{O}(n \log n)$에 계산 가능합니다.
- ✓ 전 슬라이드의 시그마 기호 안의 값은 많아야 $n-1$번 계산하므로, 문제를 제한 시간 내에 해결할 수 있게 됩니다.