

2024, 14th

Ajou Univ.

APC

Programming Contest

2024 아주대학교 프로그래밍 경시대회

Official Solutions

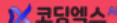
주최



주관

ANSI

후원



Division 1 – Competition Round

87명 참가

총 제출 **988** 회

맞았습니다!! **203** 회

Division 2 – Challenge Round

32명 참가

총 제출 **362** 회

맞았습니다!! **78** 회

운영

- ✓ 김유겸
- ✓ 김현빈

출제

- ✓ 김유겸^{dbrua1222} 아주대
- ✓ 김현빈^{akim9905} 아주대
- ✓ 박민석^{acepark} 아주대
- ✓ 장민우^{pani} 아주대
- ✓ 신정환^{shjohw12} 성균관대
- ✓ 이승재^{coxie} 성균관대
- ✓ 오해성^{deuslovelt} 성균관대

검수

- ✓ heeda0528 서울대
- ✓ hojoon0205 성균관대
- ✓ jk410 서울대
- ✓ lky7674 연세대
- ✓ max084 성균관대
- ✓ nick832 한양대
- ✓ palilo 성균관대
- ✓ petamingks POSTECH

Div.1	Div.2	성균관대	문제	난이도	출제자
-	A	A	이 대회는 이제 제 겁니다	Easy	김유겸 dbrua1222
A	B	B	마라탕후루	Easy	장민우 pani
B	C	-	현대모비스 특별상의 주인공은? 2	Easy	김유겸 dbrua1222
C	D	-	APC2shake!	Normal	김현빈 akim9905
D	E	C	밤양갱	Normal	김현빈 akim9905
-	F	D	열정! 열정! 열정!	Normal	박민석 acepark
E	G	E	너 재능있어	Normal	박민석 acepark
-	-	F	이상한 트리 해싱	Normal	이승재 coxie
F	H	-	지금부터 서로 죽여라	Hard	장민우 pani
-	-	G	AK47	Hard	신정환 shjohw12
G	I	-	나는 북극곰입니다	Hard	박민석 acepark
H	-	-	나는 연어입니다	Hard	김현빈 akim9905
-	-	-	최소 문자열 뽑기	Challenging	오해성 deuslovelt
I	-	-	K-mins	Challenging	김현빈 akim9905
-	-	-	마라탕후루 (hard)	Challenging	장민우 pani

-/2A/SA. 이 대회는 이제 제 겁니다

math

출제진 의도 - **Easy**

✓ 출제자: 김유겸^{dbrua1222}

-/2A/SA. 이 대회는 이제 제 겁니다

- ✓ Division 1의 상금 + shake! 상금과 Division 2의 상금 액수를 비교합니다.
- ✓ 따라서 정답은 $\max(A + C, P)$ 입니다.

1A/2B/SB. 마라탕후루

math

출제진 의도 - **Easy**

✓ 출제자: 장민우^{pani}

1A/2B/SB. 마라탕후루

- ✓ 두 가지의 풀이가 존재합니다.
- 1. $N \leq 100$ 이고, 문제에서 각 꼬치에 로봇이 과일을 꽂은 횟수가 10 000 번 이내여야 한다고 하였으므로 $100 \times 10\,000 = 10^6$ 번의 경우를 모두 확인하는 브루트포스로 해결할 수 있습니다.
- 2. 일반성을 잃지 않고 $P \geq Q$ 라고 하였을 때, $(A_i - B_i)$ 가 $(P - Q)$ 로 나누어떨어지는지 여부를 확인하면서 수학적으로 해결할 수도 있습니다.

1B/2C/- . 현대모비스 특별상의 주인공은? 2

implementation, string

출제진 의도 - **Easy**

✓ 출제자: 김유겸^{dbrua1222}

1B/2C/-. 현대모비스 특별상의 주인공은? 2

- ✓ 문제에서 제시한 것 처럼, 8방향의 **MOBIS**를 찾으면 됩니다.
- ✓ 격자의 모든 **M**부터 for문 또는 하드코딩으로 5칸을 보면 됩니다.
- ✓ 격자의 모든 **B**에서 앞 뒤 문자 2개만 보면 구현을 더 편하게 할 수 있습니다.
- ✓ 격자 테두리에 2칸짜리 빈칸을 만들어두면, 경계를 체크하는 수고를 덜어낼 수 있습니다.

1C/2D/-. APC2shake!

sort, implementation

출제진 의도 - **Normal**

✓ 출제자: 김현빈^{akim9905}

1C/2D/-. APC2shake!

- ✓ 아래의 조건을 만족시키는 10 명의 학생을 2024 APC의 성적순으로 정렬합니다.
 - 재학 중이고(jaehak),
 - ICPC에서 수상하지 못하였고(notyet),
 - shake! 최고 성적이 -1 이거나 3 보다 크거나 같은 학생
- ✓ 적절한 구조체를 활용하여 정렬할 수 있습니다.
- ✓ 시간 복잡도는 $\mathcal{O}(N \log N)$ 입니다.

1D/2E/SC. 밤양갱

greedy

출제진 의도 - **Normal**

✓ 출제자: 김현빈^{akim9905}

- ✓ $N = 1$ 인 경우를 살펴봅시다. 최소 10 초가 필요합니다.
 - `daldidalgo`
 1. `daldi`까지 입력합니다. 5 초가 필요합니다.
 2. `dal`을 복사해 붙여넣습니다. 1 초가 필요합니다.
 3. `go`를 작성합니다. 2 초가 필요합니다.
 - `daldidan`
 1. `daldida`까지 복사합니다. 1 초가 필요합니다.
 2. `n`을 입력합니다. 1 초가 필요합니다.

1D/2E/SC. 밤양갱

- ✓ $N = 2$ 인 경우를 살펴봅시다. 최소 11 초가 소요됩니다.
 1. $N = 1$ 일 때 결과를 생각하면, `daldidalgo`는 최소 8 초가 필요합니다.
 2. 동일한 내용을 복사 붙여넣기하면 1 초가 추가 소요됩니다. (`daldidalgo daldidalgo`)
 3. 여기서 `daldida`를 복사해 붙여넣고, `n`을 입력하면 2 초가 소요됩니다.

- ✓ $N = 3$ 인 경우를 살펴봅시다. 최소 11 초가 소요됩니다.
 1. $N = 1$ 일 때 결과를 생각하면, `daldidalgo`는 최소 8 초가 필요합니다.
 2. 동일한 내용을 복사 붙여넣기하면 1 초가 추가 소요됩니다. (`daldidalgo daldidalgo`)
 3. 여기서 `daldidalgo daldida`를 복사해 붙여넣고, `n`을 입력하면 2 초가 소요됩니다.

1D/2E. 밤양갱

✓ 일반화해보면,

1. $N = 2^K$ 끝인 경우

- ▶ `daldidalgo`를 최초 입력 후 K 번 붙여넣습니다. $8 + K$ 초가 소요됩니다.
- ▶ 그 다음 `daldida`를 복사해 붙여넣고, `n`을 입력하면 2초가 소요됩니다.

2. 그 외

- ▶ $2^K < N$ 을 만족하는 가장 큰 K 까지 복사 붙여넣기 합니다. $8 + K$ 초가 필요합니다.
- ▶ 나머지 `daldidalgo... daldidalgo daldida`까지 복사 붙여넣기 합니다. 1초가 필요합니다.
- ▶ `n`을 입력합니다. 1초가 필요합니다.

✓ 답은 $10 + \lfloor \log_2 N \rfloor$ 초입니다.

-/2F/SD. 련정! 련정! 련정!

ad_hoc, greedy
출제진 의도 - **Easy**

✓ 출제자: 박민석^{acepark}

-/2F/SD. 령정! 령정! 령정!

- ✓ x 의 값이 A 의 원소보다 훨씬 큰 값을 관찰할 수 있습니다.
- ✓ 배열의 앞 부분은 큰 값을 가질수록, 뒷 부분은 작은 값을 가질수록 좋습니다.
- ✓ 따라서 두 대칭되는 인덱스를 가지는 구 값 a_i 에 x 를 더하고 a_{N-i+1} 에 x 를 뺍니다.
- ✓ 여기서 i 가 증가함에 따라 조금씩 작은 값을 더해주면 됩니다.
- ✓ x 를 10^6 부터 시작하여 5 000씩 빼면서 a_i 에 x 를, a_{N-i+1} 에는 x 를 뺍니다.

-/2F/SD. 령정! 령정! 령정!

- ✓ 배열의 길이가 짝수일 경우는 모든 원소에 연산을 적용할 수 있습니다.
- ✓ 배열의 길이가 홀수일 경우 가운데 원소의 양옆에 적용한 연산의 x 값이 5 000 을 넘으면 문제 되지 않습니다.
- ✓ 따라서, $\left\lfloor \frac{N}{2} \right\rfloor$ 번 연산을 사용하면 반드시 내림차순 배열을 만들 수 있습니다.
- ✓ $\mathcal{O}(N)$ 에 문제를 풀 수 있습니다.

1E/2G. 너 재능 있어

dynamic_programming

출제진 의도 - Easy

✓ 출제자: 박민석^{acepark}

1E/2G/SE. 너 재능 있어

- ✓ DP를 이용할 생각을 해볼 수 있습니다.
- ✓ $dp(i, j) = i$ 번째 까지 이기고 j 번째까지 졌을 때 얻을 수 있는 최대 점수
 1. $dp(i, j - 1)$ 이 K 의 배수인 경우

$$dp(i, j) = \max(dp(i - 1, j) + W_{i-1}, dp(i, j - 1) - L_{j-1})$$
 2. $dp(i, j - 1)$ 이 K 의 배수가 아닌 경우

$$dp(i, j) = \max(dp(i - 1, j) + W_{i-1}, dp(i, j - 1) - \min(L_{j-1}, dp(i, j - 1) \bmod K))$$
- ✓ $\mathcal{O}(NM)$ 에 문제를 풀 수 있습니다.

-/-/SF. 이상한 트리 해싱

tree, hashing

출제진 의도 - **Hard**

✓ 출제자: 이승재^{Coxie}

-/-/SF. 이상한 트리 해싱

- ✓ 노드 두개 이하로 이루어진 트리와 동일한 해시값을 갖는 트리는 존재하지 않습니다.
- ✓ 이제 루트 노드에 무조건 자식이 있다고 가정하겠습니다.

-/-/SF. 이상한 트리 해싱

1. 트리 한 개를 만드는 과정을 살펴봅시다.
 - 루트 노드의 자식 P_1, P_2, \dots, P_n 이 있다고 해봅시다.
 - 그럼 루트의 해시값은 $2^{H(P_1)H(P_2)\dots H(P_n)}$ 입니다.
 - 따라서 $\log h = H(P_1) \times H(P_2) \times \dots \times H(P_n)$ 이고, h 가 2의 거듭제곱 수여야 합니다.
 - $H(P_i)$ 도 2의 거듭제곱이므로, $\log h$ 또한 2의 거듭제곱 입니다.
 - 루트 노드에 $\log \log h$ 의 개수만큼 자식이 있다면 해당하는 해시값을 갖습니다.

-/-/SF. 이상한 트리 해싱

2. 나머지 한 개를 만드는 상황을 생각해봅시다.

- 루트 노드에 자식이 $t = \log \log h$ 개 있는 트리와 동일한 해시값을 갖는 트리를 만들어야 합니다.
- $2 \times 2 = 2^2$ 입니다. 따라서 t 개의 노드 중 한개의 노드를 다른 노드의 자식으로 옮기면 됩니다.

1F/2H. 지금부터 서로 죽여라

case_work, game_theory

출제진 의도 - **Hard**

✓ 출제자: 장민우^{pani}

1F/2H/-. 지금부터 서로 죽여라

✓ 여러 가지 경우를 나눠서 생각해 봅시다.

1. $Y \geq A$ 나루토가 첫 턴에 공격하면 이깁니다.
2. $B \geq C$ 이고 $D \geq A$ 둘 다 회복만 하면 절대 지지 않을 수 있으므로 비깁니다.
3. $B \geq C$ 이고 $D < A$ 나루토는 무조건 지지 않을 수 있습니다.

1F/2H/-. 지금부터 서로 죽여라

3. $B \geq C$ 이고 $D < A$

- 나루토가 승리할 수 있는지만 확인하면 됩니다.
- 사스케는 절대로 회복하지 않습니다. 회복 직후 나루토가 공격시 사스케는 손해만을 보기 때문입니다.
- 이제 다음과 같은 전략을 생각해볼 수 있습니다.
 - ▶ $k(B - C) \geq C$ 가 될 때까지 나루토의 회복, 사스케의 공격을 k 번 반복합니다.
 - ▶ 이후 사스케의 공격 한번을 받아도 체력은 줄지 않습니다. 즉, 한 턴을 벌게 됩니다.
 - ▶ 따라서 $B > C$ 인 경우 나루토는 승리합니다.
 - ▶ 하지만 $B = C$ 인 경우 이 전략이 통하지 않습니다.
- 따라서 나루토와 사스케가 서로 공격만 했을 때 나루토가 더 짧은 횟수 안에 사스케를 죽일 수 있다면 승리, 아니라면 자신을 계속 회복하면서 무승부를 만들 수 있습니다.

1F/2H/-. 지금부터 서로 죽여라

4. $B < C$ 이고 $D \geq A$ 위와 반대로 해결해 줄 수 있습니다.
 5. $B < C$ 이고 $D < A$ 서로 회복하는 것이 항상 손해이므로 둘 다 공격만 합니다.
 - 공격했을 때 적을 죽이기까지의 횟수를 계산해서 누가 이기는지 판별하면 됩니다.
- ✓ 시간 복잡도는 $\mathcal{O}(1)$ 입니다.

-/-/SG. AK47

ad_hoc, binary_search

출제진 의도 - **Hard**

✓ 출제자: 신정환^{shjohw12}

-/-/SG. AK47

- ✓ 보물이 하나라면 전형적인 이분탐색 문제로, $\log N$ 번의 쿼리로 해결 가능합니다.
- ✓ 이 문제는 어떻게 해결할 수 있을까요? 일단 $\left[1, \frac{N}{2}\right]$ 에 쿼리를 날려봅시다.
- ✓ 이때 보물이 하나 있다고 응답한다면, $\left[1, \frac{N}{2}\right]$, $\left[\frac{N}{2} + 1, N\right]$ 각각에 보물이 하나씩 있다는 것이므로 각 구간에서 이분탐색을 하여 해결 가능합니다.
- ✓ 그렇지 않다면, $\left[1, \frac{N}{2}\right]$ 에 두 보물이 있거나 $\left[\frac{N}{2} + 1, N\right]$ 에 두 보물이 있다는 의미입니다.

- ✓ 이때 $\left[1, \frac{N}{2}\right], \left[\frac{N}{2} + 1, N\right]$ 각각에서 위 행위를 반복한다면,
최악의 경우에 N 번 이상의 쿼리를 날려야합니다.
- ✓ 이런 식의 방법으로는 문제를 해결하기 어려워보입니다.

- ✓ 두 보물이 인접해 있을 때 보물의 위치를 어떻게 알 수 있을까요?
- ✓ 다른 모든 위치에 보물이 없음이 확인된 게 아니라면, 보물이 a 번째, $a + 1$ 번째 위치에 존재할 때 $[*, a]$ 또는 $[a + 1, *]$ 에 쿼리를 날려야합니다.
- ✓ 쿼리를 나누지 말고 길이가 고정된 쿼리를 sliding window의 방식으로 밀면서 생각해볼까요?

-/-/SG. AK47

- ✓ 길이 K 로 고정된 쿼리를 앞에서부터 순차적으로 날린다고 생각해봅시다.
- ✓ $[l, l + K - 1]$ 에서 최초로 보물이 하나 있다고 응답한다면, 두 가지 케이스로 나눕니다.
 1. 이전 쿼리에서는 두 보물이 있던 경우
 - ▶ $l - 1$ 에 보물이 있고, $[l, K]$ 에 보물이 하나 있음
 2. 이전 쿼리에서는 보물이 없던 경우
 - ▶ $l + k - 1$ 에 보물이 있고, $[l + k, N]$ 에 보물이 하나 있음

- ✓ 그렇다면 퀴리의 길이는 어떻게 설정해야 할까요?
- ✓ 길이가 너무 짧다면 많은 퀴리를 날려야 하고,
너무 길다면 중간에 보물이 모여있는 경우를 처리하기 어렵습니다.
- ✓ 중간에 보물이 모여있는 경우에 대해 처리 가능하면서 가장 긴 길이는 절반입니다.
- ✓ 길이를 절반으로 설정하여 위 방법을 시행한다면 $\left\lfloor \frac{N+1}{2} \right\rfloor + 1$ 번의 퀴리로 문제를 해결할 수 있습니다.

1G/2I/-. 나는 북극곰입니다

dijkstra, binary_search

출제진 의도 - **Hard**

✓ 출제자: 박민석^{acepark}

1G/2I/- . 나는 북극곰입니다

1. 이분 탐색을 이용한 풀이

- x 분에 출발하여 도착할 수 있다면, $x - 1$ 분에 출발하여도 도착할 수 있습니다.
- 따라서 이분 탐색과 다익스트라를 활용할 생각을 해볼 수 있습니다.
- 북극곰이 T 초 후 출발할 때를 기준으로,
 N 번 빙하에 도착할 수 있는 최단 경로를 다익스트라를 통해 구합니다.
- 시간 복잡도는 $O(N \log M \log T)$ 입니다.

2. 역방향에서 시작하는 다익스트라 알고리즘을 이용한 풀이

- N 번 노드부터 출발합니다.
- 우선순위 큐에 (현재 빙하에 머무를 수 있는 최대 시간, 현재 노드) 를 관리합니다.
- $\min(t - d, \text{현재 빙하에 머무를 수 있는 최대 시간} - d)$ 와 비교하면서 다익스트라를 수행합니다.
- 이 경우 시간복잡도는 $O(N \log M)$ 입니다.

1H/-/- . 나는 연어입니다

sorting, dsu, graph_theory, binary_search

출제진 의도 - **Hard**

✓ 출제자: 김현빈^{akim9905}

1H/-/. 나는 연어입니다

- ✓ 유의미한 구간의 개수는 $\mathcal{O}(M)$ 개입니다.
- ✓ 어떤 두 구간 $[l_1, r_1]$ 과 $[l_2, r_2]$ 를 생각해 봅시다.
- ✓ 이때 집합 $\{l_1, r_1 + 1, l_2, r_2 + 1\}$ 를 정렬한 결과가 $[a, b, c, d]$ 라고 해봅시다. ($a \leq b \leq c \leq d$)
- ✓ 그러면 $[1, a)$ 사이의 크기를 가진 연어는 모두 같은 결과를 가지게 됩니다.
- ✓ 마찬가지로 $[a, b), [b, c), [c, d), [d, 10^9]$ 에서 같은 구간에 속한 크기를 가진 연어는 같은 결과를 가집니다.

1H/-/. 나는 연어입니다

- ✓ 따라서 우리는 $\mathcal{O}(M)$ 개의 구간에 대해서 N 번까지 갈 수 있는지 확인하면 됩니다.
- ✓ N 번까지 갈 수 있는지의 여부는 분리집합이나 DFS/BFS 등을 이용하여 구현할 수 있습니다.
- ✓ 각 구간에 속하는 연어의 수는 이분 탐색을 통해 찾을 수 있습니다.
- ✓ 시간 복잡도는 $\mathcal{O}(M^2 + N \log N)$ 입니다.

-/-/SH. 최소 문자열 뽑기

stack, greedy

출제진 의도 - **Hard**

✓ 출제자: 오해성^{deuslovelt}

-/-/SH. 최소 문자열 뽑기

1. 행이 한개일 때 문제를 해결해봅시다.

- 논리적인 스택을 구현한 후 열을 훑는다고 생각해봅시다.
 - ▶ 스택에 첫째 열의 알파벳을 담고 다음열로 넘어갑니다.
 - ▶ 스택의 맨 위 알파벳이 사전순으로 현재 열의 알파벳보다 크고 열을 가릴 수 있는 횟수가 남아있다면 스택의 알파벳에 해당하는 열을 무조건 가리는 것이 이득입니다.
 - ▶ 따라서 현재 열의 알파벳보다 스택의 알파벳이 큰 동시에 열을 가릴 수 있는 횟수가 남아있다면 계속해서 스택에서 가릴 열을 지정하는 그리디한 풀이를 낼 수 있습니다.

-/-/SH. 최소 문자열 뽑기

2. 행이 여러개일 때 문제를 해결해봅시다.

- 두 가지 관찰을 할 수 있습니다.

2.1 배열의 첫째 행부터 행이 증가하는 대로 문제를 해결하여도 상관이 없다.

2.2 배열의 행이 여러개로 늘어나면서 생기는 차이는 스택에 존재하는 현재 열보다 크기가 큰 알파벳을 모두 가릴 수 없을 때 발생한다.

- 첫째 행부터 문제를 푼다 하였을때 스택에 첫째 열의 알파벳을 담고 다음 열로 넘어갑니다.
- 스택의 맨 위 알파벳이 사전순으로 현재 열의 알파벳보다 크고 해당 열을 가릴 수 있다면 무조건 가리는 것이 이득입니다.

-/-/SH. 최소 문자열 뽑기

- 스택에 현재 열의 알파벳 보다 사전순으로 큰 알파벳이 있는 경우
 - ▶ 행이 한 개일 때와 달리 두가지 경우로 문제가 나뉘게 됩니다.
- 스택 내부에 현재 열의 알파벳보다 사전순으로 큰 알파벳을 모두 가릴 수 있는 경우
 - ▶ 행이 한 개일 때와 같이 그리디하게 모두 가리는 것이 무조건 이득입니다.
- 만약 전부 다 가리지 못한다면 그중에 가릴 수 있는 만큼 가려야합니다.
- 이는 아래 행으로 넘어가서 해당 열들에 대해서만 본 문제와 같은 풀이를 재귀적으로 하면 됩니다.

11/-/- . K-mins

sort, set/map, segment_tree, math, combinatorics

출제진 의도 – **Challenging**

✓ 출제자: 김현빈^{akim9905}

- ✓ Naïve하게 접근한다면, $\mathcal{O}(N^2K)$ 로 시간초과를 받습니다.
- ✓ K 번째 수를 고정하고 생각할 수 있습니다.
- ✓ (A_i, i) 의 값을 오름차순으로 정렬한다면, A_i 이전에 있는 값들은 모두 A_i 보다 작은 값을 가집니다.
- ✓ 이때 A_i 보다 작은 값들의 인덱스를 오름차순으로 담고 있는 set 을 준비합니다.

11/-/. K-mins

- ✓ 이제 `set`의 원소 중에서 i 보다 작은 값을 가지는 왼쪽 끝 $left$ 를 지정한 뒤, A_i 가 구간의 K 번째 원소가 될 수 있는 오른쪽 끝 $right$ 를 지정해줍니다.
- ✓ 각각의 왼쪽 끝에 대하여 오른쪽 끝으로 살펴봐야하는 값의 최대 개수는 K 개입니다.
- ✓ 그럼 하나의 인덱스 i 에 대하여 $A_i \times (right - left)$ 만큼의 값이 정답에 더해집니다.
- ✓ 시간 복잡도는 $\mathcal{O}(NK \log N)$ 입니다.
- ✓ 번외로, 출제자의 첫 풀이는 `min`, `max` 값을 저장하는 세그먼트 트리를 이용한 풀이였습니다.

-/-/-. 마라탕후루 (hard)

number_theory, extended_euclid, math

출제진 의도 - **Challenging**

- ✓ 제출 -번, 정답 -명 (정답률 -%)
- ✓ 처음 푼 사람: -, -분
- ✓ 출제자: 장민우^{pani}

-/-/. 마라탕후루(hard)

- ✓ 일반성을 잃지 않고 $P \geq Q$ 라고 합시다.
- ✓ $A_i + P \times x_i = B_i + Q \times y_i$ 를 만족하는 가장 작은 음이 아닌 정수쌍 (x_i, y_i) 들을 구합니다.
- ✓ 이때 만족하는 (x_i, y_i) 쌍을 구할 수 없다면 답은 NO입니다.

-/-/. 마라탕후루(hard)

- ✓ 모든 i 에 대해서 $A_i = B_i$ 라면,

A_i 에 P 를 $\frac{Q}{\gcd(P, Q)}$ 번 더한 것과 B_i 에 Q 를 $\frac{P}{\gcd(P, Q)}$ 번 더한 것이 같습니다.

- ✓ 이것이 P 와 Q 를 각각 더 더해줄 수 있는 최솟값입니다.

- ✓ 따라서 $\sum_{k=1}^N x_i$ 와 $\sum_{k=1}^N y_i$ 의 값의 차이와

위에서 구한 더해줄 수 있는 최솟값의 차이와의 관계를 통해 원하는 답을 구해낼 수 있습니다.

- ✓ 시간 복잡도는 $\mathcal{O}(N \log X)$ 입니다.