

2025 상반기 전남대학교 PIMM 알고리즘 파티

Solutions Commentary Editorial

전남대학교 게임개발동아리 PIMM 알고리즘 연구회

This Contest is operated by

고민규_{jjkmk1013}

김근성_{onsbtyd}

박종현_{belline0124}

이윤수_{lys9546}

정영도_{0do}

최정환_{jh01533}

dongwook7

lycoris1600

realpsdoingdamyoo

sjhi00

tony9402

utilforever

with .

추첨상

- 부정행위를 한 것이 확실한 참가자가 제외되었습니다.

자세한 내용은 [에디토리얼 리포지토리](#)에 함께 게시된 <대회 제출 소스코드 AI 사용 평가 보고서>를 참조해주시요.

버거킹 와퍼 세트 문제 해결 수의 세제곱을 가중치로 5인 선정

- kwoncycle
- dk10211
- ckj31110
- golazcc83
- fermion5

아이스 카페 아메리카노 T 문제 해결 수의 제곱을 가중치로 5인 선정

- luciaholic
- oh040411
- nemomaru
- ssjjss
- asker5325

아이스 카페 아메리카노 T 문제 해결 수를 가중치로 5인 선정

- patata22
- minpro0818
- starboard
- ssafyiwan
- eka

아이스 카페 아메리카노 T 5인 선정

- tph01198
- didtldms2525
- aru0504
- nabina1395
- tkvl94

참가 기념 solved.ac 프로필 배경 / 뱃지

- 부정행위를 한 것이 확실한 참가자가 제외되었습니다.

자세한 내용은 [에디토리얼 리포지토리](#)에 함께 게시된 <대회 제출 소스코드 AI 사용 평가 보고서>를 참조해주시요.



프로필 배경: 보금자리 바깥으로
2025학년도 전기 학위수여식 동아리 현수막 배경
© 정희수



프로필 뱃지: 케이크 한 조각
아망추 팀, <대학카페키우기> 게임 에셋
© 안상이

문제	의도한 난이도	출제자
A 치매예방수칙 3.3.3	Easy	이윤수 ^{lys9546}
B 비장의 일격 (Small)	Normal	이윤수 ^{lys9546}
C 내 이름 나무	Hard	김근성 ^{onsbtyd}
D 블록 굴리기	Hard	고민규 ^{jjkmk1013}
E 쌓기나무	Hard	정영도 ^{0do}
F Min Max Mex	Hard	최정환 ^{jh01533}
G 보물 찾기	Challenging	최정환 ^{jh01533}
H 비장의 일격 (Large)	Challenging	이윤수 ^{lys9546} , 최정환 ^{jh01533}

A. 치매예방수칙 3.3.3

#string, #parsing

출제진 의도 — **Easy**

- 출제자: 이윤수_{lys9546}
- 제출 237회, 정답 186건
정답률: 79.747%
- 최초 해결: 14cheung, +1분

A. 치매예방수칙 3.3.3

- ✓ 문제가 요구하는 내용을 그대로 구현합니다.
- ✓ 주어진 문자열을 ., |, :, # 으로 분리하여 수를 추출한 후, 수를 모두 더합니다.

B. 비장의 일격 (Small)

#implementation, #bruteforcing

출제진 의도 — **Normal**

- 출제자: 이윤수_{lys9546}
- 제출 326회, 정답 102건
정답률: 33.436%
- 최초 해결: 14cheung, +5분

B. 비장의 일격 (Small)

- ✓ $S \leq 100, K = 2$ 이므로, 비장의 일격을 모든 쌍에 대해서 시도하여도 시간 제한을 통과할 수 있습니다.
- ✓ 모든 쌍에 대하여 비장의 일격을 시도할 때, $O(N^2)$ 의 시간 복잡도를 가지므로 2회 ($K = 2$) 시도하는 데 $O(N^4)$ 의 시간 복잡도를 가집니다.

C. 내 이름 나무

#graph, #bfs, #multisource_bfs, #shortest_path

출제진 의도 — **Hard**

- 출제자: 김근성^{onsbtyd}
- 제출 142회, 정답 26건
정답률: 19.014%
- 최초 해결: sadtreap, +8분

C. 내 이름 나무

문제 요구사항

- ✓ K 거리 이내로 인접한 같은 이름 쌍이 있는지 확인합니다.

BFS/DFS로 풀기

- ✓ 이름이 붙은 모든 노드에서 BFS 혹은 DFS를 시도하여 K 거리 이내에 같은 이름이 있는지 확인합니다.
- ✓ 이름이 N 개 존재하는 경우, $O(N^2)$ 의 시간 복잡도로 시간 초과가 발생합니다.
- ✓ 같은 이름에 대해 중복되어 탐색되므로 탐색 수를 줄여야합니다.

C. 내 이름 나무

멀티소스 BFS로 풀기

- ✓ 같은 이름에 대해 한 번에 탐색하면 탐색 수를 줄일 수 있습니다.
 - ✓ 중복을 제외하고 등장 가능한 이름은 최대 1,000가지이므로 멀티소스 BFS를 수행할 수 있습니다.
1. 임의의 노드 A 에서 인접한 노드를 탐색하면서, 이미 방문 처리가 되어있으면서 시작점이 다른 노드 B 를 만나야 합니다.
이 때 가장 가까운 이름 쌍의 거리 = A 에 기록된 거리 + B 에 기록된 거리 + 1 입니다.
 2. 가장 가까운 이름 쌍의 거리 $\leq K$ 라면 POWERFUL CODING JungHwan, 아니라면 so sad를 출력합니다.

C. 내 이름 나무

- ✓ 시간 복잡도는 $O(\text{이름 가짓수} \times N)$ 입니다.
- ✓ 원래 정환이의 이름은 “최 대값을구하는맥스함수”였습니다.

D. 블록 굴리기

#implementation, #dp

출제진 의도 — **Hard**

- 출제자: 고민규^{jjkmk1013}
- 제출 57회, 정답 16건
정답률: 28.07%
- 최초 해결: 14cheung, +27분

D. 블록 굴리기

문제 요구사항

- ✓ 정확히 K 회 이동하여 정해진 목표 타일에 도착할 수 있는 시작점의 개수를 구합니다.

문제 풀이 아이디어

- ✓ 블록의 이동은 가역적이므로 출발점과 도착점을 바꿀 수 있습니다.
- ✓ 도착점에서 목표 타일에서 정확히 K 번 이동하여, 출발점에 세워진 상태로 도달할 수 있는 모든 일반 타일의 개수를 구하는 문제와 동일합니다.

D. 블록 굴리기

DP로 풀기

- ✓ $DP_{i,j,k,d} = k$ 회 이동하여 (i, j) 의 타일에 블록이 d 상태로 도달할 수 있는지 여부
- ✓ 블록의 상태를 3개 혹은 5개 유형으로 표현할 수 있습니다.
 - 1개: 블록이 세워져있음
 - 2개 혹은 4개: 블록이 눕혀져있음 (눕혀진 방향에 따라 개수가 다름)

D. 블록 굴리기

- ✓ 시간복잡도는 $O(NMK)$ 입니다.
- ✓ 제한 조건에 따라, 목표 타일에는 블록을 배치할 수 없습니다.
만약 K 회 이동하여 목표 타일에 돌아올 수 있다면, 이 경우를 제외하여야 합니다.

E. 쌓기나무

#implementation

출제진 의도 — **Hard**

- 출제자: 정영도^{0do}
- 제출 44회, 정답 16건
정답률: 36.364%
- 최초 해결: sorohue, +67분

E. 쌓기나무

$M = 1$ 인 경우

- ✓ $N = 200,000, Q = 500,000$
- ✓ 각 쿼리를 최대 $O(\log N)$ 내에 처리해야 합니다.
- ✓ FRONT 쿼리는 STACK 쿼리 수를 출력하므로 $O(1)$ 입니다.
- ✓ SIDE 쿼리는 $M = 1$ 일 때 쌓기나무 전체의 최대 높이입니다. 이분탐색 혹은 세그먼트 트리를 사용하여 $O(\log N)$ 에 저장과 검색을 모두 수행합니다.
- ✓ TOP 쿼리는 맵을 사용하여 $O(1)$ 에 처리합니다.
- ✓ STACK / REMOVE 쿼리를 처리하면 1만큼 블록의 높이가 변화하므로, 높이 정보를 맵으로 관리합니다. 최대 높이가 갱신될 때 인접 쌓기나무의 높이를 탐색하여 자료의 저장과 검색을 $O(1)$ 에 처리합니다.

E. 쌓기나무

$M = 200,000$ 인 경우

- ✓ FRONT 쿼리는 정면에서 각 좌표마다 보이는 최대 높이의 합입니다.
 - ✓ 맵에서 자료의 저장과 검색을 $O(1)$ 에 처리하므로, 높이 정보를 담는 맵을 N 개 생성하여 $O(N + Q)$ 에 처리합니다.
- ✓ SIDE 쿼리는 FRONT와 같은 전략으로 $O(M + Q)$ 에 처리합니다.
- ✓ TOP 쿼리는 맵을 사용하여 $O(1)$ 에 처리합니다.

E. 쌓기나무

- ✓ $\langle M = 1 \text{인 경우} \rangle$ 풀이의 시간복잡도는 $O(Q)$ 입니다.
- ✓ $\langle M = 200,000 \text{인 경우} \rangle$ 풀이의 시간복잡도는 $O(N + M + Q)$ 입니다. 이분 탐색 등의 추가적인 최적화 대책을 사용하여 시간복잡도 $O(\log NQ + \log MQ)$ 가 가능할 수 있습니다.

F. Min Max Mex

#greedy

출제진 의도 — **Hard**

- 출제자: 최정환^{jh01533}
- 제출 136회, 정답 33건
정답률: 24.265%
- 최초 해결: sorohue, +7분

F. Min Max Mex

문제 요구사항

- ✓ K 개의 원소를 제거하여 없앨 수 있는 가장 작은 음이 아닌 정수를 구합니다.

Min Mex

- ✓ 배열의 각 원소의 개수를 셉니다.
- ✓ 0에서부터 각 원소에 대해, 원소의 개수가 K 보다 작거나 같은지 확인합니다.

Max Mex

1. 배열을 정렬합니다.
2. 배열의 끝에 입력 조건 바깥의 큰 수 INF를 삽입합니다.
3. 배열을 순회하면서 각 원소 사이에 빈 구간이 있으면 그리디하게 값을 채웁니다.

G. 보물 찾기

#math, #constructive, #interactive

출제진 의도 — **Challenging**

- 출제자: 최정환^{jh01533}
- 제출 50회, 정답 5건
정답률: 10%
- 최초 해결: fermion5, +276분

G. 보물 찾기

X 를 찾지 않는 방법

- ✓ 맨해튼 거리가 같은 두 지점의 쪽지는 X 를 XOR해도 서로 같습니다.
- ✓ 맨해튼 거리가 다른 두 지점의 쪽지는 X 를 XOR해도 다릅니다.
- ✓ N 이 크다면 이분탐색이나 이와 비슷한 기법을 이용하여 \log 스케일에 해결할 수 있습니다.

G. 보물 찾기

<X를 찾지 않는 방법> 예시

1. 아래와 같이 땅을 판다면, 남는 후보는 두 개 입니다.

```
.....  
.@....  
..@...  
...@..  
....@.  
.....
```

2. N 이 짝수일 때, 남는 후보는 한 개입니다.

```
@.@.@.  
.....  
.....  
.....@  
.....  
.....@
```

G. 보물 찾기

3. N 이 홀수일 때, 남은 후보는 두 개입니다.

.....
.@.@.@.
.....
.....@.
.....
.....@.
.....

G. 보물 찾기

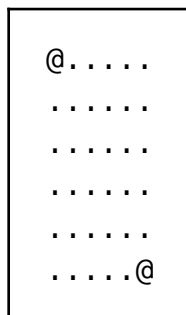
X 를 찾는 방법

- ✓ X 를 알고 있다면, 세 건의 질문으로 문제를 해결할 수 있습니다.

.../..
\./...
.X....
/.\...
...\..
@...\@

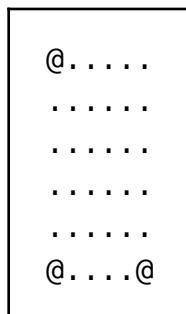
- ✓ 남는 후보는 1개입니다.

G. 보물 찾기



- ✓ 이 질문의 답을 A, B 라고 하고 X 가 적용되지 않았을 때 받아야 하는 답을 a, b 라고 할 때, 다음의 수식이 성립합니다.
 - ✓ $a + b = 2N - 2$
 - ✓ $(A \oplus X) + (B \oplus X) = 2N - 2$
- ✓ 가능한 X 의 개수는 $N/2$ 보다 작습니다.

G. 보물 찾기



✓ 이와 같이 질문한 뒤, X 후보의 땅을 모두 파서 해결할 수 있습니다.

G. 보물 찾기

더 생각해보기

- ✓ 상수 P 회의 질문만을 사용하여 X 를 정확하게 구할 수 있을까요?

H. 비장의 일격 (Large)

#dp

출제진 의도 — **Challenging**

- 출제자: 이윤수^{lys9546}, 최정환^{jh01533}
- 제출 7회, 정답 28건
정답률: 25%
- 최초 해결: sorohue, +60분

H. 비장의 일격 (Large)

- ✓ 성질 1. 비장의 일격을 맞은 두 문자와 그 사이의 문자열을 구간이라고 할 때, 각 구간은 서로 겹치지 않고 포함관계를 가지지 않는 것 중 최적의 존재합니다.
- ✓ 성질 2. 같은 알파벳에 한 번의 비장의 일격만 시행하는 최적의 방법이 항상 존재합니다.

H. 비장의 일격 (Large)

<성질 1. 비장의 일격을 맞은 두 문자와 그 사이의 문자열을 구간이라고 할 때, 각 구간은 서로 겹치지 않고 포함관계를 가지지 않는 것 중 최적이 존재합니다.>

- ✓ 만약 어떤 구간 A 에 다른 구간 B 가 존재한다면, B 구간에 대해 비장의 일격을 시행하지 않아도 결과는 동일합니다. 즉, 포함 관계를 가지지 않는 것 중 최적이 존재합니다.

H. 비장의 일격 (Large)

- ✓ $DP_{i,j} = \max(i \text{까지 확인하였고, 비장의 일격을 } j \text{회 시행할 수 있을 때 없앨 수 있는 구간 길이})$
- ✓ 전이식
 - ✓ i 번째 문자를 포함하는 비장의 일격을 시행할 때, $S_{i-k} = S_i$ 를 만족하는 모든 K 에 대해 $\max(DP_{i-K,j-1})$
 - ✓ i 번째 문자를 포함하지 않는 비장의 일격을 시행할 때, $DP_{i-1,j}$
- ✓ 이 계산은 $O(N^2K)$ 의 시간 복잡도를 요구하나, DP를 하나 더 도입하여 $O(NK)$ 로 최적화할 수 있습니다.
- ✓ $DP'_{c,j} = \max(\text{임의의 인덱스 } i \text{에 대해, } S_i \text{가 } C \text{이고 } i-1 \text{까지 } j \text{회의 비장의 일격을 시행했을 때, 없앨 수 있는 글자의 최댓값에 } n-i \text{를 더한 값})$

H. 비장의 일격 (Large)

<성질 2. 같은 알파벳에 한 번의 비장의 일격만 시행하는 최적의 방법이 항상 존재합니다.>

$\dots[a\dots a]\dots[a\dots a]\dots$

상황 예시: $[a\dots a]$ 에 각각 비장의 일격을 시행하는 것 보다 양 끝에 시행하는 것이 더 낫습니다.

- ✓ 같은 알파벳 4개에 비장의 일격을 2회 시행하는 대신, 양 끝의 알파벳 2개에 비장의 일격을 1회 시행하는 것이 더 최적입니다.
- ✓ 시간복잡도는 $O(NL)$ ($L = X$ 를 제외한 알파벳의 개수 ≤ 25)입니다.