제 9회

인하대학교 프로그래밍 경진대회

대회 결과 및 해설

주최

인하대학교 컴퓨터공학과
Department of Computer Engineering

인하대학교
SW중심대학사업단

주관

후원

A. 모비스

의도한 난이도 : **EASY** #구현

#출제 : 황진익

제출 104번, 정답 68명 (정답률 65.38%)

처음 푼 사람 : **최준형**, 1분

IUPC

A. 모비스

- 출제 의도
  - 문자열을 다룰 수 있는가?

- 주어진 문자열에 'M', 'O', 'B', 'I', 'S'가 모두 존재하는지 판단하는 문제

- 반복문 등으로 간단하게 구현할 수 있다

IUPC

B. 스파이

의도한 난이도 : **EASY** #재귀 #완전탐색

#출제 : 김민겸

제출 80번, 정답 50명 (정답률 62.50%)

처음 푼 사람 : **정상준**, 11분

IUPC

B. 스파이

- 출제 의도
  - 모든 경우의 수를 검사하는 완전 탐색 알고리즘을 구현할 수 있는가?

- 민겸이는 임무를 N일간 수행한다

- 민겸이는 하루에 6가지 행동 중 하나를 선택하여 할 수 있다

- 따라서 민겸이가 N일간 임무를 수행하는 방법은 6^N가지

IUPC

B. 스파이

- N이 최대 8이므로, 최악의 경우 임무를 수행하는 방법은 6^8 = 1,679,616 가지
  - 최악의 경우에도 모든 경우를 시간 안에 탐색할 수 있다
- 재귀 함수를 이용해 (임무를 진행한 일수, 가장 최근 임무를 진행한 장소) 를 인자로 두어 구현할 수 있다
- 재귀 함수 대신 8중 for문 등으로도 해결할 수 있다
- 만약 N이 훨씬 크다면 어떻게 문제를 풀 수 있을까요?

IUPC

C. 멀티탭 충분하다

의도한 난이도 : **HARD** #정렬 #애드혹

#출제 : 김현민

제출 22번, 정답 5명 (정답률 22.72%)

처음 푼 사람 : **김동현**, 70분

IUPC

C. 멀티탭 충분하다

- 출제 의도
  - 이 문제에 적용되는 특별한 성질을 찾아 풀이할 수 있는가?

IUPC

C. 멀티탭 충분하다

- $N < 3$ 인 경우는 자명하므로 고려하지 않는다

- i번 멀티탭과 j번 멀티탭의 콘센트가 1개 이상 겹치는 경우
  - 둘의 각도가 다른 경우 : 1개 겹친다
  - 둘의 각도가 같은 경우 : 최소 1개, 최대 min(Ai, Aj)개 겹친다

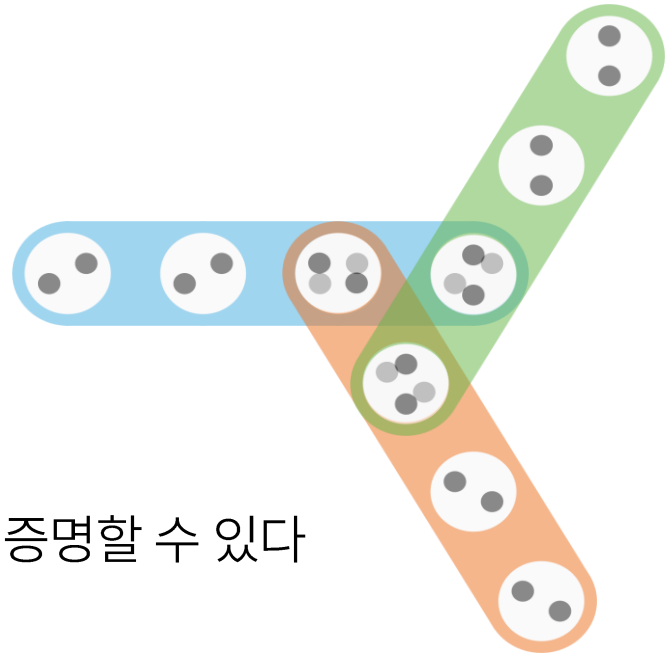
- f층의 멀티탭과 g층의 멀티탭의 각도가 같으려면,
  - $f \equiv g \pmod 3$ 을 만족해야 한다

  - 1, 4, 7, ... 층
    - 가장 다수의 멀티탭이 여기에 속한다
  - 2, 5, 8, ... 층
    - 둘째로 다수의 멀티탭이 여기에 속한다
  - 3, 6, 9, ... 층
    - 가장 소수의 멀티탭이 여기에 속한다

IUPC

C. 멀티탭 충분하다

- 1, 4, 7, ... 층
  - 제일 긴 멀티탭부터 여기에 둔다
- 2, 5, 8, ... 층
  - 위에서 사용하고 남은 멀티탭 중 제일 긴 것부터 여기에 둔다
- 3, 6, 9, ... 층
  - 남은 것을 여기에 둔다



- 이렇게 배치하면 →
  - 각 그룹에서 가장 긴 것만 보이며
  - 가장 적은 수의 콘센트가 보이게 할 수 있다
- 위와 같이 3개의 멀티탭이 보이게 하는 것이 최선임을 증명할 수 있다

IUPC

C. 멀티탭 충분하다

- 답 : $A_a + A_b + A_c - 3$
  - A는 멀티탭의 콘센트 수를 내림차순으로 정렬한 수열
  - $a = 1$
  - $b = \begin{cases} \left\lfloor \frac{N}{3} \right\rfloor + 1, & \text{if } N \equiv 0 \text{ (MOD 3)}. \\ \left\lfloor \frac{N}{3} \right\rfloor + 2, & \text{otherwise.} \end{cases}$
  - $c = \begin{cases} b + \left\lfloor \frac{N}{3} \right\rfloor + 1, & \text{if } N \equiv 2 \text{ (MOD 3)}. \\ b + \left\lfloor \frac{N}{3} \right\rfloor, & \text{otherwise.} \end{cases}$

IUPC

D. 사탕 팔찌

의도한 난이도 : **HARD** #BFS

#출제 : 정치훈

제출 17번, 정답 0명 (정답률 0.00%)

처음 푼 사람 : -, NaN분

IUPC

D. 사탕 팔찌

- 출제 의도
  - 올솔브 방지

IUPC

D. 사탕 팔찌

- 정해
- 가능한 사탕 묶음들을 그래프의 정점으로, 이을 수 있는 두 사탕 묶음을 방향이 있는 간선으로 이어 봅시다.
- 정점이 O(N!)개인 그래프가 나옵니다.
- 그러면 이 그래프에서 모든 정점을 한 번 씩만 지나는 해밀턴 회로를 구하는 문제가 됩니다.
- 하지만 일반적인 그래프에서 해밀턴 회로를 구하는 문제는 NP-Hard 문제라서 매우 어렵습니다. 어떻게 해야 할까요?

- 비슷한 유형의 문제 중 오일러 회로 문제가 있습니다.
- 오일러 회로 문제는 모든 정점이 아니라, 모든 간선을 한 번 씩만 지나는 회로를 찾는 문제입니다.

IUPC

D. 사탕 팔찌

- 처음에 만든 그래프의 정점을 간선으로 바꿀 수 있을까요?
- 놀랍게도 가능합니다!

- 예를 들어 다음과 같은 그래프가 있다고 생각해 봅시다.
- ABCD – BCDE – CDEF
- 정점 ABCD와 BCDE의 공통 문자열은 BCD, 정점 BCDE와 CDEF의 공통 문자열은 CDE 입니다.

- 이 때, 두 간선 BCD와 CDE 둘과 동시에 연결되어 있는 정점은 BCDE가 유일하다는 사실을 알 수 있습니다!
- 따라서 (K-1)개의 문자열을 사용하여 나올 수 있는 모든 부분 사탕 묶음을 정점으로 하고, 두 정점과 동시에 연결되어 있는 사탕 묶음을 간선으로 이어 새로운 그래프를 만들 수 있습니다.

- 이제 문제가 오일러 회로 문제를 푸는 문제로 바뀌었으니, O(N+M) DFS를 이용하여 풀 수 있습니다.

IUPC

E. 중력 큐

의도한 난이도 : **NORMAL** #덱 #구현

#출제 : 안용모

제출 244번, 정답 23명 (정답률 9.42%)

처음 푼 사람 : **송현성**, 82분

IUPC

E. 중력 큐

- 출제 의도
  - 지문에서 주어진 내용을 구현하기 위해 적절한 자료구조를 사용할 수 있는가?

- 큐의 뒤가 아래쪽을 향하도록 놓여진 경우 큐의 뒤쪽에서 pop 연산이 이루어진다
- 양 방향으로 push와 pop이 가능한 자료 구조인 덱을 사용한다

- rotate 쿼리 이후 큐가 세로 방향일 때 큐에서 빠져나오는 공에 대해 고려한다
- count 쿼리의 경우 push나 pop이 이루어 질 때마다 값을 갱신한다

IUPC

F. 배 옮기기

의도한 난이도 : **HARD** #수학

#출제 : 정치훈

제출 15번, 정답 2명 (정답률 13.33%)

처음 푼 사람 : **송현성**, 147분

IUPC

F. 배 옮기기

- 출제 의도
  - 주어진 상황을 그래프화 할 수 있고, 최단 경로 알고리즘을 적용할 수 있는가?

- 배가 최대 15척이므로, 배가 위치해 있는 모든 상황을 그래프의 정점으로, 배를 이동하여 정점 사이에 이동할 수 있는 경우를 간선으로 하는 그래프를 만듭시다.
- 간선의 가중치는 배를 옮겼을 때 걸리는 시간입니다.

- 정점이 O(2^N) 개인 그래프가 나옵니다.
- 가중치 있는 그래프에서 최단경로를 구하는 문제가 됩니다.
- 다익스트라 알고리즘을 이용하면 됩니다

IUPC

G. 인경호의 나무

의도한 난이도 : **NORMAL** #DFS #조합론

#출제 : 김현민

제출 10번, 정답 2명 (정답률 20.00%)

처음 푼 사람 : **최준형**, 83분

IUPC

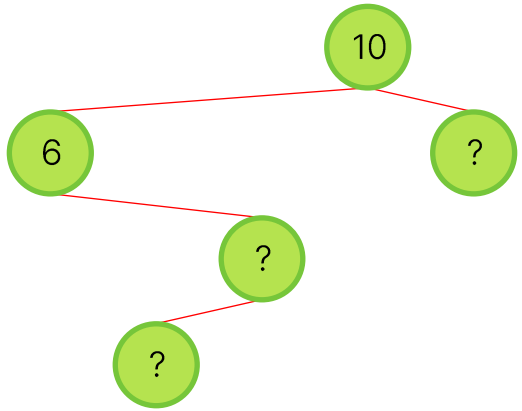
G. 인경호의 나무

- 출제 의도
  - 이진탐색트리의 성질을 알고 있고, 조합을 사용하여 경우의 수를 계산할 수 있는가?

IUPC

G. 인경호의 나무

- 인경호의 나무 : 이진탐색트리
- 이진탐색트리는 중위순회하면 오름차순의 수열이 된다



$M = 15$

→ 6 ? ? 10 ? → 6 ? ? 10 ?

$\binom{3}{2}$ $\binom{5}{1}$

중복된 키를 허용하지 않으므로, 조합을 이용해 구하면 된다.

IUPC

H. 직사각형 피자

의도한 난이도 : **NORMAL** #이분탐색

#출제 : 안용모

제출 347번, 정답 19명 (정답률 5.47%)

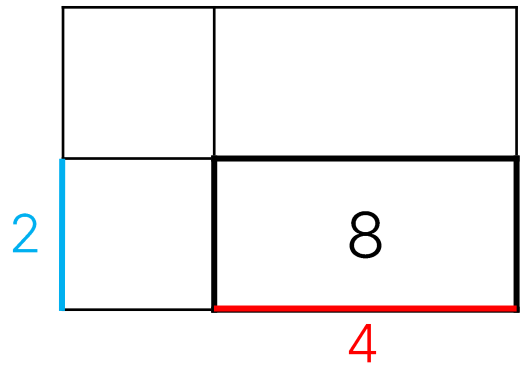
처음 푼 사람 : **최준형**, 19분

IUPC

- 출제 의도
  - 이분탐색을 활용하여 문제를 해결 할 수 있는가?

H. 직사각형 피자

- 피자 조각의 개수가 최대 (100,001)^2개이다
- 피자 조각의 크기를 모두 저장하고 탐색할 수 없다



$$\overline{4} \times \overline{2} = 8$$

- 피자 조각 하나의 넓이는 커팅으로 인해 생긴 가로 방향의 선분 하나의 길이와 세로 방향의 선분 하나의 길이의 곱이다

IUPC

H. 직사각형 피자

- 가로 방향과 세로 방향의 선분들의 길이를 각각 저장한다
- 한 방향의 선분 하나에 대해 다른 방향의 선분에서 길이의 곱이 K 이하인 선분들의 개수를 찾을 수 있고, 이는 한 방향의 선분 하나로 만들 수 있는 조각들 중에서 넓이가 K 이하인 조각의 개수와 같다

- 다른 방향의 선분들의 길이를 정렬한다면 한 방향의 선분에 대해 이분 탐색으로 길이의 곱이 K 이하인 선분들의 개수를 $O(logN)$이나 $O(logM)$에 구할 수 있다.

- 이를 한 방향의 선분 전체에 대해 반복하면 모든 조각에 대해 넓이가 K 이하인 조각의 개수를 $O(MlogN)$이나 $O(NlogM)$에 구할 수 있다

IUPC

I. 기계오리 연구

의도한 난이도 : **HARD** #DP

#출제 : 안용모

제출 149번, 정답 6명 (정답률 4.02%)

처음 푼 사람 : **정상준**, 81분

IUPC

출제 의도

Knapsack DP를 응용하여 문제를 해결할 수 있는가?

IUPC

I. 기계오리 연구

- 기계오리에 대해 각 배터리의 장착 여부만 고려하면 된다
- 기계오리에 장착되는 배터리의 개수를 최소화해야 한다

→ 0-1 Knapsack 문제의 변형

- 기계오리에 장착된 배터리의 전력량의 최댓값은 50,000이다

→ 1차원 배열로 각 전력량에 대한 정보를 저장할 수 있다

- 기계오리에 장착된 배터리의 전력량의 합이 i일 때,
- 기계오리에 장착된 배터리의 종류의 최솟값을 $dp[i]$라고 하자
  - $dp[0] := 0$, $dp[i] := \infty$ $(i \neq 0)$로 초기화한다

IUPC

- 첫번째 배터리부터 각 배터리마다 모든 전력량에 대해 dp의 값을 갱신하면 된다
  - 이때, 배터리 하나는 한번만 장착할 수 있으므로 큰 전력량부터 값을 갱신한다
- j번째 배터리에 대해 값을 갱신하는 경우 점화식은 다음과 같다
  - $dp[i + I_j] := \min(dp[i + I_j], dp[i] + 1)$

- 모든 배터리에 대해 dp의 값을 갱신한 뒤, 0을 제외하고 $dp[i]$의 값이 K 이하인 전력량 i의 수와 그 값들을 각각 출력하면 된다

J. 마왕의 성

의도한 난이도 : **HARD** #분리집합

#출제 : 안용모

제출 46번, 정답 3명 (정답률 6.52%)

처음 푼 사람 : **정상준**, 77분

IUPC

J. 마왕의 성

- 출제 의도
  - Disjoint Set을 사용하여 문제를 해결할 수 있는가?

IUPC

J. 마왕의 성

- 마왕의 성의 위치가 정해졌을 때 마왕이 걷을 수 있는 세금의 합의 최댓값은
- 영토로 만들 수 있는 부지를 모두 영토로 포함한 경우 걷는 세금의 합이다

- 마왕의 성의 위치가 정해졌을 때 영토로 만들 수 있는 부지는
  - 높이가 마왕의 성의 높이 이하이며
  - 마왕의 성의 높이 이하인 부지만을 지나 마왕의 성으로 도달할 수 있는 부지이다

- 각 위치에 대해 그래프 탐색 등의 방법으로 도달 가능한 부지를 모두 찾는 경우
- 최악의 경우의 시간 복잡도는 $O(N^2M^2)$ → 시간 초과

IUPC

J. 마왕의 성

- 마왕의 성의 위치가 정해졌을 때 성보다 높은 위치에 있는 부지는 고려할 필요가 없다
- 따라서 각 부지의 위치를 높이 순서대로 오름차순으로 탐색할 수 있다

- 높이가 h인 부지들에 대해 탐색하는 경우
  - 현재 부지에서 상하좌우로 인접하고 높이가 h 이하인 부지로 이동할 수 있다
  - 현재 부지와 높이가 h 이하인 인접한 부지에서 이동 가능한 부지로 이동할 수 있다

- 높이가 h 이하인 부지들에 대해 먼저 이동 가능한 부지를 구했기 때문에, 높이가 h인 부지의 경우 상하좌우로 인접한 부지에 대해서만 고려하면 된다

IUPC

J. 마왕의 성

- 분리 집합을 이용해 이를 구현할 수 있다

- 높이가 현재 부지보다 작거나 같은 인접한 부지에 대해 union한다
  - 이때 union할 때 root 노드의 번호와 같은 집합에 있는 세금의 합을 갱신한다
- 한 높이의 부지들을 모두 탐색했을 때, 그 높이의 부지들에 대해 root의 세금의 합이 그 위치에 성을 세운 경우 얻을 수 있는 세금의 합의 최댓값이다
  - 성을 지은 부지보다 높이 있는 부지를 거치지 않고 현재 부지에서 방문할 수 있는 모든 부지를 union했기 때문이다

- 모든 부지에 대해 높이 오름차순으로 위 작업을 수행하여 모든 위치에 각각 마왕의 성을 세우는 경우 얻을 수 있는 세금의 합의 최댓값을 $O(NM)$에 구할 수 있다
  - 상수가 크므로 실제 실행 시간은 일반적인 $O(NM)$ 알고리즘보다 느리다

IUPC