2023 UDPC

Official Solutions

문제		의도한 난이도	출제자
A	탁구 경기	**Easy**	한동규queued_q
B	UDPC 파티	**Easy**	노지훈g3gogogo
C	카드 뒤집기	**Medium**	김범수qjatn0120baek
D	동전 퍼즐	**Medium**	한동규queued_q
E	현대모비스 입사 프로젝트	**Medium**	김도훈99asdfg
F	햄버거최대 몇개드실수있나요?	**Hard**	노지훈g3gogogo
G	윤이는 엄청난 것을 훔쳐갔습니다	**Hard**	한동규queued_q
H	인덕션	**Hard**	노지훈g3gogogo
I	Quartet	**Challenging**	박신욱bnb2011
J	벚꽃 엔딩	**Challenging**	장래오leo020630

A. 탁구 경기

`implementation`
출제진 의도 – **Easy**

- ✓ 통계:
  - Junior: 제출 22번, 정답 17번, 정답률 77.273%
  - Senior: 제출 49번, 정답 29번, 정답률 59.184%
- ✓ First Solve:
  - Junior: **Beishekeev Daniyar**[UNIST], 2분
  - Senior: **김경민**[POSTECH], 2분
- ✓ 출제자: 한동규[queued_q]

A. 탁구 경기

- 문제에서 주어진 대로 점수를 시뮬레이션하면 됩니다.
- 변수 X 와 Y 를 정의합니다.
- 반복문을 사용하여 N 번의 라운드를 확인하며 X 와 Y 에 점수를 더해 주다가, 조건문을 사용하여 $\text{abs}(X - Y) \geq 2$ 가 되면 `break` 로 반복문을 빠져나옵니다.
- 마지막으로 `X:Y` 형식에 맞게 출력합니다.

B. UDPC 파티

`math`

출제진 의도 – **Easy**

- ✓ 통계:
  - Junior: 제출 103번, 정답 13번, 정답률 12.621%
  - Senior: 제출 120번, 정답 28번, 정답률 23.333%
- ✓ First Solve:
  - Junior: **Amirbek Abdirasulov**[UNIST], 10분
  - Senior: **Mansur Mukimbekov**[UNIST], 8분
- ✓ 출제자: 노지훈[g3gogogo]

B. UDPC 파티

- ✓ 이 문제에서 서로 바뀔 수 있는 U와 C, 그리고 D와 P는 같은 문자로 취급됩니다.
- ✓ 그렇기 때문에 다음 두 가지 성질을 만족합니다.
  - 달구와 포닉스는 둘 다 선정될 수 있거나 선정될 수 없습니다. 즉, D와 P는 둘 다 출력하거나, 출력하지 않습니다.
  - 투표 결과인 문자열이 비어 있는 경우가 없으므로 달구와 포닉스가 선정될 수 없는 경우에는 무조건 윤이가 선정될 수 있습니다. 즉, C를 출력하는 경우는 없습니다.
- ✓ 따라서 투표 결과에서 U와 C의 개수 UC, D와 P의 개수 DP를 통해 누가 제일 귀여운 마스코트로 선정될 수 있는지 알아낼 수 있습니다.

B. UDPC 파티

- 한 마스코트가 제일 귀여운 마스코트로 선정되기 위해서는 그 마스코트가 받은 투표수가 다른 두 마스코트가 각각 받은 투표수보다 많아야 합니다.
- 그렇기 때문에 각 마스코트가 선정되려면 다음 조건을 만족해야 합니다.
  - 윤이: $(DP + 1)/2 < UC$
  - 달구, 포닉스: $0 < DP$
- 위 조건을 만족하는 모든 마스코트의 알파벳을 순서대로 출력하면 됩니다.

C. 카드 뒤집기

`constructive`
출제진 의도 – **Medium**

- 통계:
  - Junior: 제출 37번, 정답 16번, 정답률 43.243%
  - Senior: 제출 47번, 정답 25번, 정답률 53.191%
- First Solve:
  - Junior: **Beishekeev Daniyar**[UNIST], 22분
  - Senior: **Mansur Mukimbekov**[UNIST], 16분
- 출제자: 김범수[`qjatn0120baek`]

C. 카드 뒤집기

- 먼저 N 이 적혀져 있는 카드를 뒤집은 후에는 다른 카드를 뒤집을 수 없습니다.
- 따라서 N 이 적혀진 카드를 가장 마지막으로 뒤집어야 합니다.
- 이제 나머지 카드들을 적절하게 배열해야 합니다.

C. 카드 뒤집기

- ✓ 일단 $N-1$이 적혀있는 카드는 맨 끝에 두어야 합니다. 그렇지 않으면 $N-1$이 적혀있는 카드를 뒤집은 후에 다른 카드를 뒤집을 수 없기 때문입니다.
- ✓ $N-1$이 적혀있는 카드를 뒤집었으니 이제 뒤집어야 하는 카드는 $N-1$장입니다. 그런데 이후에 $N-2$가 적혀져 있는 카드를 뒤집지 않으면 이후에는 $N-2$라는 카드를 뒤집을 수 없습니다. 따라서 $N-1$을 뒤집은 후에는 $N-2$라는 카드를 뒤집어야 합니다.

C. 카드 뒤집기

- 이 과정을 계속해서 반복하면 아래와 같이 두 종류의 배열이 가능합니다.
  - $[\ldots, 6, 4, 2, N, 1, 3, 5, \ldots]$
  - $[\ldots, 5, 3, 1, N, 2, 4, 6, \ldots]$

  이때 카드는 $N-1, N-2, \ldots, 2, 1, N$ 순서로 뒤집어야 합니다.

D. 동전 퍼즐

`brute_force`
출제진 의도 – **Medium**

- 통계:
  - Junior: 제출 15번, 정답 10번, 정답률 66.667%
  - Senior: 제출 35번, 정답 19번, 정답률 54.286%
- First Solve:
  - Junior: **Beishekeev Daniyar**[UNIST], 29분
  - Senior: **Mansur Mukimbekov**[UNIST], 24분
- 출제자: 한동규[queued_q]

D. 동전 퍼즐

- 최종 배치를 어디에 만들지 찾는 것이 관건입니다.
- (a, b) 만큼 평행이동한 위치에 최종 배치를 만들 때 옮겨야 하는 동전의 개수는, 현재 배치에 있는 동전들 중 최종 배치에 놓여 있지 않은 동전의 수와 같습니다.
- a와 b를 각각 -10부터 10까지 바꿔 가면서, 옮겨야 하는 동전의 개수를 세었을 때 최소가 되는 (a, b) 쌍을 찾으면 됩니다.
- 위와 같이 구현하면 4중 반복문을 쓰는 코드가 나옵니다.
- 인덱스가 배열 범위 밖을 벗어나는 경우 처리에 주의합시다.

E. 현대모비스 입사 프로젝트

`greedy`, `sorting`
출제진 의도 – **Medium**

- 통계:
  - Junior: 제출 16번, 정답 9번, 정답률 56.250%
  - Senior: 제출 41번, 정답 24번, 정답률 58.537%
- First Solve:
  - Junior: **Beishekeev Daniyar**[UNIST], 34분
  - Senior: **오상규**[DGIST], 22분
- 출제자: 김도훈[99asdfg]

E. 현대모비스 입사 프로젝트

- ✓ N 개의 강의 중 K 개를 골랐을 때, 두 종류의 역량 합의 최대를 찾아야 합니다.
- ✓ 당연하게도, N 개의 강의 중 K 개를 고르는 경우의 수는 ${}_NC_K$ 이므로, 가능한 모든 경우의 수를 찾아보는 방식으로는 문제를 해결할 수 없습니다.
- ✓ 그러나, 3 개의 종류 중 강화시키고자하는 역량 2 개를 고르는 경우는 고작 3 가지만 있습니다.
- ✓ 또한, 강화하고자 하는 2 개의 역량을 이미 고른 시점에서, 두 역량의 합의 최대는 단순히 해당하는 두 역량의 합이 최대인 K 개의 강의를 고르는 것으로 구할 수 있습니다.
- ✓ 따라서, 강화할 역량을 고르는 세 가지 경우에 대해 각각 합의 최대를 구해주면 시간복잡도 $\mathcal{O}(N \log N)$ 안에 문제를 해결할 수 있습니다.

F. 햄버거최대 몇개드실수있나요?

`greedy`, `sorting`
출제진 의도 – **Hard**

- ✓ 통계:
  - Junior: 제출 17번, 정답 4번, 정답률 23.529%
  - Senior: 제출 58번, 정답 14번, 정답률 24.138%
- ✓ First Solve:
  - Junior: **Amirbek Abdirasulov**[UNIST], 72분
  - Senior: **Mansur Mukimbekov**[UNIST], 38분
- ✓ 출제자: 노지훈[g3gogogo]

F. 햄버거최대 몇개드실수있나요?

- ✓ 콜라 효과가 많이 중첩될수록, 그리고 먹은 햄버거의 질량이 클수록 햄버거를 먹어 증가하는 질량이 많이 감소합니다. 즉, 위 속 질량이 적게 증가합니다.
- ✓ 따라서 콜라 효과가 많이 중첩된 때에 질량이 큰 햄버거를 먹어야 위 속 질량이 최소가 되고, 이를 이용해 위의 용량의 최솟값을 구할 수 있습니다.
- ✓ 햄버거를 먹는 순서는 상관없기 때문에 $1, 2, ..., N$ 번째로 햄버거를 먹을 때의 각 콜라 효과의 중첩 횟수를 정렬한 배열을 $Count$, 햄버거의 질량을 정렬한 배열을 $Mass$ 라 할 때, 위의 용량의 최솟값은 아래의 식으로 계산할 수 있습니다.

$$\sum_{i=0}^{N} \frac{Mass[i]}{2^{Count[i]}}$$

- ✓ 이때, 두 배열을 정렬한 기준은 오름차순이나 내림차순으로 같아야 합니다.

F. 햄버거최대 몇개드실수있나요?

- ✓ 각 콜라에 대해 콜라 효과의 중첩 횟수를 일일이 L 번 더하면 시간 복잡도가 $\mathcal{O}(KL)$ 이 되기 때문에 시간 초과가 발생합니다.
- ✓ 따라서 prefix sum과 같은 방식을 활용해 연산 횟수를 줄여줘야 합니다.
- ✓ 예를 들어, 콜라를 마시는 시기를 정렬하고, 인덱스 i 를 1 부터 N 까지 증가시키면서 현재 인덱스에서 콜라를 마시는 횟수만큼 현재 인덱스에서의 콜라 효과의 중첩 횟수인 CNT 를 증가시키고, 콜라 효과가 끝나는 인덱스인 $i+L$ 를 저장하고, 현재 인덱스에서 콜라 효과가 끝나는 횟수만큼 CNT 를 감소시키고, $Count[i] = CNT$ 를 하면 $\mathcal{O}(K+L)$ 로 해결할 수 있습니다.

G. 윤이는 엄청난 것을 훔쳐갔습니다

`graph_traversal`
출제진 의도 – **Hard**

- ✓ 통계:
  - Junior: 제출 9번, 정답 6번, 정답률 66.667%
  - Senior: 제출 37번, 정답 13번, 정답률 35.135%
- ✓ First Solve:
  - Junior: **정윤제**POSTECH, 43분
  - Senior: **Mansur Mukimbekov**UNIST, 46분
- ✓ 출제자: 한동규`queued_q`

G. 윤이는 엄청난 것을 훔쳐갔습니다

- ✓ 어떤 리프 노드에 대해 (윤이와의 거리) < (경찰과의 거리) 이면, 윤이는 해당 리프 노드를 향해 일직선으로 달려서 탈출할 수 있습니다.
- ✓ 윤이, 달구, 포닉스 각각으로부터 DFS 또는 BFS를 실행하여 각 리프 노드까지의 거리 d_a, d_b, d_c를 구합니다.
- ✓ 모든 리프 노드 중 $d_a(i) < d_b(i)$ 이면서 $d_a(i) < d_c(i)$ 인 리프 노드 i가 존재하면 윤이는 탈출할 수 있고, 그런 리프 노드가 존재하지 않으면 윤이는 탈출할 수 없습니다.
- ✓ 시간복잡도는 $\mathcal{O}(N)$ 입니다.

H. 인덕션

DP

출제진 의도 – **Hard**

- 통계:
  - Junior: 제출 19번, 정답 4번, 정답률 21.053%
  - Senior: 제출 24번, 정답 8번, 정답률 33.333%
- First Solve:
  - Junior: **Adil Sultanov**[UNIST], 88분
  - Senior: **Mansur Mukimbekov**[UNIST], 67분
- 출제자: 노지훈[g3gogogo]

H. 인덕션

- 인덕션에는 요리할 수 있는 공간이 세 개 있으므로, 각 음식을 요리하는 방법은 총 세 가지입니다.
  - 1) 첫 번째 공간을 필요한 온도로 설정하고 음식을 요리
  - 2) 두 번째 공간을 필요한 온도로 설정하고 음식을 요리
  - 3) 세 번째 공간을 필요한 온도로 설정하고 음식을 요리
- 그리고 요리한 음식의 개수와 세 공간의 온도를 인덱스로 하는 배열 DP를 설정할 수 있습니다.
- $DP[i][t_a][t_b][t_c]$: 지금까지 요리한 음식의 개수가 i개, 현재 세 공간의 온도가 순서대로 t_a, t_b, t_c일 때, 지금까지 버튼을 눌러야 하는 최소 횟수

✓ 그러면 $i+1$ 번째 음식을 요리하기 위해 필요한 온도가 t_{i+1} 일 때, 아래와 같이 점화식을 세울 수 있습니다.

- 1) $DP[i+1][t_{i+1}][t_b][t_c]$
  $= min(DP[i+1][t_{i+1}][t_b][t_c], DP[i][t_a][t_b][t_c] + min((t_a - t_{i+1})\%10, (t_{i+1} - t_a)\%10))$

- 2) $DP[i+1][t_a][t_{i+1}][t_c]$
  $= min(DP[i+1][t_a][t_{i+1}][t_c], DP[i][t_a][t_b][t_c] + min((t_b - t_{i+1})\%10, (t_{i+1} - t_b)\%10))$

- 3) $DP[i+1][t_a][t_b][t_{i+1}]$
  $= min(DP[i+1][t_a][t_b][t_{i+1}], DP[i][t_a][t_b][t_c] + min((t_c - t_{i+1})\%10, (t_{i+1} - t_c)\%10))$

H. 인덕션

- 버튼을 눌러야 하는 최소 횟수를 찾아야 하므로 배열 DP를 처음에 $25\,000$ 이상의 값으로 초기화해야 합니다.
- 지문의 네 번째 규칙에 따라 온도는 0부터 9에서 순환하는 구조를 가지기 때문에, 한 음식을 요리하기 위해 버튼을 누르는 횟수는 5회를 넘지 않습니다. $(5 \times 5\,000 = 25\,000)$
- 점화식의 초기 상황인 $DP[0][0][0][0]$에 0을 넣고, $DP[0][0][0][0]$부터 $DP[N][9][9][9]$까지 초깃값이 아닌 인덱스에 대하여 위 점화식을 따라 진행합니다.
- 그러면 $DP[N][0-9][0-9][0-9]$에서 최솟값이 모든 음식을 순서대로 요리하기 위해 버튼을 눌러야 하는 횟수의 최솟값이 되고, $\mathcal{O}\left(1\,000N\right)$으로 해결할 수 있습니다.

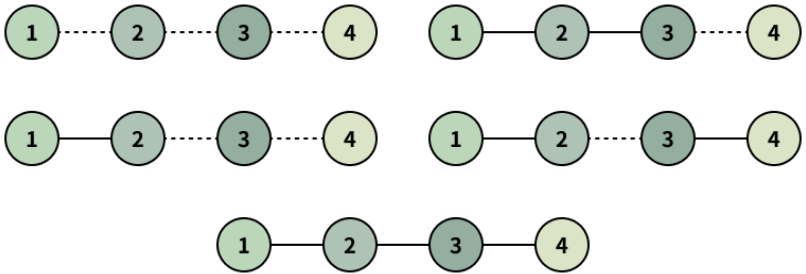
I. Quartet

graphs, ad hoc
출제진 의도 – **Challenging**

- ✓ 통계:
  - Junior: 제출 19번, 정답 0번, 정답률 0.000%
  - Senior: 제출 32번, 정답 4번, 정답률 12.500%
- ✓ First Solve:
  - Junior: -
  - Senior: **Mansur Mukimbekov**[UNIST], 117분
- ✓ 출제자: 박신욱[bnb2011]

- 네 명의 학생을 자리가 인접한 순서로 선발한다고 생각해 봅시다.
- 학생을 정점으로 하고, 두 학생 간의 시너지를 간선으로 하는 그래프를 구성합니다. 간선에는 해당 시너지를 만족시켰을 때 얻을 수 있는 조화도를 가중치로 둡니다.

✓ 인접한 자리 배치에서 조화도를 얻을 수 있는 경우는 아래와 같이 다섯 가지가 있습니다.



- 이 중 위의 네 가지 경우는 조화도가 가장 큰 시너지를 최대 두 개까지 선택하는 것으로 묶어서 생각할 수 있습니다.
- 결국 문제에서 따져봐야 할 경우는 아래의 두 가지입니다.
  - 가중치가 가장 큰 순서대로 최대 두 개의 간선 고르기
  - 사이클이 아니면서 세 개의 간선을 지나는 단순 경로 중, 간선의 가중치 합이 가장 큰 경로 찾기

- 첫 번째 경우는 쉽습니다. 입력받은 가중치를 오름차순으로 정렬해서 구하거나, 아니면 가중치를 두 번 순회해서 구할 수 있습니다. $O(M)$에 해결 가능합니다.
- 두 번째 경우는 조금 어렵습니다. 주어지는 그래프가 밀집되어 있는 경우에는 세 개의 간선을 거쳐가는 경로의 개수가 대략 $O(N^3)$개 정도로 매우 많아질 수 있기 때문에, 단순히 모든 조합을 따져보는 방식으로는 문제를 해결할 수 없습니다.

- 세 개의 간선 중 중간에 위치한 간선을 고정해봅시다.
- 이렇게 간선을 고정하고 나면, 이제 그 간선의 양쪽 정점에 연결된 다른 간선을 하나씩 선택해 경로를 구성할 수 있습니다.
- 여기서 구하고자 하는 경로는 **가중치 합이 가장 큰** 경로이므로, 그 정점에 연결된 모든 간선들을 순회할 필요는 없습니다. 가장 가중치가 큰 몇 개의 간선만 확인해보는 것으로 충분합니다.
- 선택한 간선들끼리 사이클을 이뤄선 안되고, 이미 고정한 간선을 또 사용하는 경우도 배제해야 합니다. 따라서 가중치가 가장 큰 순서대로 **세 개**의 간선만 각각 확인해보면 됩니다.

- ✓ 정점마다 가장 가중치가 큰 간선 세 개를 찾는 과정은 다 합해서 $\mathcal{O}(N+M)$에 가능합니다. 그리고 중간 간선을 고정하는 경우에 대해 양쪽 간선을 선택하는 과정도 $\mathcal{O}(1)$에 가능하니, 전체 문제를 $\mathcal{O}(N+M)$에 해결할 수 있습니다.

J. 벚꽃 엔딩

`two_pointer`, `data_structure`
출제진 의도 – **Challenging**

- ✓ 통계:
  - Junior: 제출 0번, 정답 0번, 정답률 0.000%
  - Senior: 제출 19번, 정답 2번, 정답률 10.526%
- ✓ First Solve:
  - Junior: -
  - Senior: **Almas Abilkhanov**[UNIST], 162분
- ✓ 출제자: 장래오[leo020630]

J. 벚꽃 엔딩

- 연속된 벚꽃의 시작 지점을 고정해봅시다. 이를 x 라 합시다.
- x 번 나무로부터 번호가 증가하는 방향으로 나무를 추가하면, 특정 지점까지는 나무들의 개화 기간이 겹치지만 해당 지점 이후로는 겹치는 기간이 없어집니다. 이 지점을 D_x 라 합시다.
- 그리고, $i < j$ 이면 $D_i < D_j$ 입니다.

- ✓ 따라서 이를 투 포인터로 관리해줄 수 있습니다.
- ✓ 개화 기간이 겹치는지는 현재 살펴보고 있는 구간에 속한 나무의 시작 정점과 끝 정점을 set 등의 자료구조에 저장한 후, $max(S_i) \leq min(E_i)$ 인지를 확인해주면 됩니다.
- ✓ 투 포인터가 아닌 파라메트릭 서치와 슬라이딩 윈도우를 사용하는 방법이나, 세그먼트 트리 등의 자료구조를 사용해도 문제를 해결할 수 있습니다.