교내 프로그래밍 대회
10회차 리뷰

알고리즘 소모임 씨앗

주의사항

1. 어디까지나 제 해결법을 제시합니다.
judge에서 전부 패스받는 것은 확인했지만
더 좋은 해결방법이 존재할 수 있습니다.

2. 파이썬으로 설명드립니다.
다만, 알고리즘과 자료구조를 중점으로 설명드리겠습니다.

3. 모든 코드는 github에서 확인 가능합니다.
https://github.com/jm-kor-00/BaekjoonStudy/tree/master/Contest/KoreatechContest_10

문제해결에 접근하는 순서

1. 문제 조건 및 해결 방법 구상
2. 전수조사로 해결가능한 문제인가? (브루트포스)
3. 그렇다면 어떤 알고리즘을 적용해야 할까? (그리디, DP)
4. 자료구조를 어떻게 사용할 것인가?
5. 구현

A. 10주년 기념 카드 준비

상자에 적힌 숫자를
~~1000, 900, 500, 400,~~ 100, 90, 50, 40, 10, … 순으로 나머지 연산
(100이 큰 숫자들의 공약수이므로)

$$F(X) = ((((X \% 100) \% 90) \% 50) \% 40) / 10$$

#수학 #구현

B. 우주에서 온 시그널

델타파와 알파파를 최소한으로 수정해서 아래의 조건 만족.
델타 OR 알파 = 우주에서 날아온 신호

1. 내가 비트 연산을 안다.
2. 난 비트 연산 처음 들어봤다. 이진수는 다행히 안다.

핵심 :
bit-or의 결과가 0 이려면 델타-알파 모두 0임.
bit-or의 결과 1 이려면 둘 중 하나만 1이면 됨.

B-2. 비트 연산 모른다.

입력이 예를 들어 A, B, C = 13, 2, 11 이라면.

정수를 이진수 리스트로 변환하는 함수를 작성하여 다음과 같이 수정
arrA = [1,1,0,1] , arrB = [0,0,1,0], arrC = [1,0,1,1]

리스트의 길이는 일정하게 맞춰줘야 함 (정수니까 32자리?)
각 자리를 비교하여 필요한 수정횟수를 구하면 됨.

#구현 #비트연산

C. 나는야 외주왕!

핵심 포인트 :

1. 외주를 반드시 순서대로 마감해야 한다는 점.
2. 친구에게 부탁하는 경우는 시간제한이 없다는 점.
3. 시간이 오래 걸리는 작업들을 최대한 맡기고 남은 작업들만 영훈이가 직접 했을 때 몇 개의 작업을 할 수 있는가?

C. 나는야 외주왕!

핵심 알고리즘

1. 새로운 작업 Wn이 들어오면 친구들에게 맡긴 작업들과 비교

2-1. Wn보다 빨리 끝나는 작업이 없다면 영훈이가 해야 함.

2-2. Wn보다 빨리 끝나는 작업을 영훈이가 하고 Wn은 부탁함.

3. 영훈이가 더 이상 작업을 받을 수 없으면 알고리즘 종료.

C. 나는야 외주왕!

최대값, 최소값이 연속적으로 필요한 문제는
힙으로 구현한 우선순위 큐의 사용을 항상 염두해야 함.
삽입 enque : O(log N), 삭제 deque : O(logN)

문제C를 구현하기 위해선 우선순위 큐의 사용이 필수적(아닌가?)

#자료구조 #우선순위 큐 #구현
추천 예제 : 11279(S2) ,1715(G4)

D. KoreaTech 최고의 사랑 노선을 찾아

그래프 탐색 문제라는 것은 눈치 챌 수 있음. 그런데
주어진 자료를 어떻게 효율적으로 탐색할 것인가?

핵심 포인트 :
1. 주어진 자료(노선, 정류장) 을 어떻게 저장하고 가공할 것인가
2. 탐색 중에 최단 거리를 어떻게 반환할 것인가

D. koreaTech 최고의 사랑 노선을 찾아

핵심 아이디어1. 딕셔너리(map) 을 이용한 자료 정리

정류장 번호가 몇 번인지 모르기때문에
List[정류장] = [노선1, 노선2] 와 같은 방식은 어려움

따라서 {key : value} 형태의 자료구조를 사용하는 편이 유리함.

D. koreaTech 최고의 사랑 노선을 찾아

핵심 아이디어2. 최단 거리 반환 방법

그래프 탐색 과정 중에 목적지에 도착하면
최단 거리를 반환해야 하므로 BFS를 통해 가까운 곳부터 탐색.

큐에 넣을 때,
현재 정류장 번호와 현재까지의 이동거리를 함께 삽입해서
BFS를 진행하고, 목적지 도착시에는 (이동거리+1) 반환

D. koreaTech 최고의 사랑 노선을 찾아

대부분의 그래프 탐색 문제는 BFS, DFS로 해결할 수 있지만
단순한 인접행렬, 인접리스트의 형태로는 접근하기
어려운 문제들도 많음.

또, 탐색 과정에 이동 거리나 그 밖의 정보들이
함께 갱신되어야 하는 경우도 많음.

#그래프탐색 #MAP
추천 예제 : 9375(S3), 2206(G3)

E. 우주 개척

E번 역시, 그래프 탐색입니다.

각 노드(행성) 간의 인접여부를 판단하고
방향 그래프를 생성하고, 그래프를 탐색합니다.

때로는 인접여부가 입력 자체에서 주어지는 게 아니라,
별도의 방식으로 구해야 하는 경우도 있다는 것을
염두해야 합니다.
#그래프이론 #너비우선탐색

F. 마을을 지켜라

아주 좋은 문제가 출제됐다고 생각합니다.
많은 학생들(저 포함)이 그리디로 낚였기 때문에..
하지만 다시보면 전형적인 DP문제라는 것을 알 수 있습니다.

핵심 포인트:

1. 점화식 도출하기
2. 메모지에이션을 어떻게 구현할 것인가?

F. 마을을 지켜라

이 문제의 경우,
i번부터 j번 집을 가지고 마을과 마왕이 싸웠을 때
마을이 정신력을 얼마나 더 많이 갖는지를 F(i, j) 로 두면
다음과 같은 점화식이 세워집니다.
F(i, j) = town[i] − F(i+1, j) or town[j] − F(i, j−1)

마을이 승리하려면 F(i,j) 가 최대한 커져야 하니까
둘 중 큰 값으로 취해야 합니다.

F. 마을을 지켜라

DP table	0	5	0	9	2
0					
5					
0					
9					
2					

F(i, j) =
town[i] – F(i+1, j)
or
town[j] – F(i, j-1)

F. 마을을 지켜라

DP 자체에 대한 설명은 과거 강의 내용을 첨부합니다.

https://velog.io/@jm-kor-00/%EC%94%A8%EC%95%97-1%EC%A3%BC%EC%B0%A8-DP

다차원의 테이블이 필요하거나, 자료구조 개념이 혼용되는

DP 문제들은 까다롭기 때문에 많은 연습이 필요.

#다이나믹 프로그래밍 #DP

추천 예제 : 2193(S3), 11057(S1), 2629(G3)

G. 색칠하기

핵심 포인트.
만약 마지막에 칠한 것이 세로, 빨간색이라면
가로로 완벽한 검은색 줄은 존재할 수 없다.(반대의 경우도 마찬가지)

따라서 완벽하게 유지된 줄을 찾으면 그 줄의 색이 정답.

대충 구현하면 되지만,
2가지 색에 대해 모두 탐색하거나
모든 행렬에 대해 탐색하는 것은 비효율적. (정답이긴 하겠지만)

H. KoreaTech 경제 연구소

핵심 포인트
1. 어떤 구간 S,H (시작: S, 종료: S+H, 높이: H) 에 대해서
겹치는 구간의 높이가 이미 n 이면, 높이는 n + H가 됨.

2. 반드시 그 구간에 속하지 않더라도
기존에 더 큰 값이 있다면 출력은 그 값이 되어야 함.

3. 입력된 순서대로 값을 갱신하며
결과를 출력한다는 것이 포인트.

H. KoreaTech 경제 연구소

n번째 입력쌍(시작,종료,높이) = S, E, H 에 대해서

앞에서 처리한 구간들(n-1번째까지) 중에서
겹치는 구간(n번째가 그 위로 올라가는 경우) 중,
가장 높은 곳의 높이를 p_H 라고 하면

n번째 입력쌍은 (S, E, H + p_H) 로 새롭게 저장함.

결과에는 H + p_H 와 기존 최대 높이 중 큰 값을 추가하고
최대높이가 바뀌면 갱신함.

H. KoreaTech 경제 연구소

#누적 합 #구현

추천 예제 : 2571(G3)

I. 코너 주문 대기열의 고통

주어진 입력(시간제한, 코너별 시간)을 바탕으로 학생들의 주문을 모두 처리할 수 있는 최소 시간을 구해야 함.

핵심 포인트

1. 어떤 시간 N에 모든 주문을 처리할 수 있는 지 확인.
2. 브루트 포스로 접근하기엔 탐색해야 할 값이 너무 많음
   -> 어떻게 탐색해야 할까?

I. 코너 주문 대기열의 고통

어떤 범위안에서 적합한 값을 찾는 가장 효율적인 방법
→ 이분탐색

이분탐색으로 값의 범위를 줄여가며 가장 적합한 값을 찾는다.

#이분탐색 #매개 변수 탐색
추천 예제 : 2110(G3), 2805(S2), 1300(G1,쉬움),1561(G2)

J. 서버 대여 일정 짜기

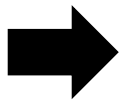
(시작시간, 반납시간, 대여개수) 순서쌍이 주어질 때,
각 시간대별 서버 대여 수를 출력하는 문제

핵심 포인트.
1. 서버 대여 수가 바뀌는 시간을 어떻게 확인하고, 어떻게 출력하는가
2. 어떤 자료구조를 사용해야 하는 가?
(단순히 배열에 넣으면 안됨, 시험횟수 X 시간 범위 = 10^9)

J. 서버 대여 일정 짜기

1. 서버 대여 수가 바뀌는 시간을 확인하는 방법

2 ~ 5	4 ~ 7	7 ~ 9
6대	2대	4대

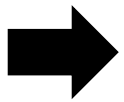


2	3	4	5	6	7	8	9

J. 서버 대여 일정 짜기

1. 서버 대여 수가 바뀌는 시간을 확인하는 방법

4 ~ 7	2 ~ 5	7 ~ 9
2대	2대	4대



2	3	4	5	6	7	8	9

2	4	5	7	9
+6	+2	-6	-2 + 4	-4

낭비되는 메모리를 줄이고
탐색 속도를 위해서
Map(Dictionry) 를 사용함

J. 서버 대여 일정 짜기

2. 어떻게 결과를 출력할까

2	4	5	7	9
6	2	-6	2	-4

순서대로 key:Value 를 꺼내오기 위해
우선순위 큐를 사용할 수 있음.
정렬할 수도 있음. 둘 다 O(N log N)

시작	종료	서버 개 수

J. 서버 대여 일정 짜기

#누적 합 #구현

추천 예제 : 20440(G3)

https://www.acmicpc.net/problem/20440