2023 UDPC

Official Solutions

Problem		Difficulty intention	Problem maker
A	Table Tennis	**Easy**	queued_q
B	UDPC party	**Easy**	g3gogogo
C	Card Flip	**Medium**	qjatn0120baek
D	Coin Puzzle	**Medium**	queued_q
E	Hyundai Mobis Joining Project	**Medium**	99asdfg
F	How-many-burgers can you eat at once?	**Hard**	g3gogogo
G	Yunee Stole the Precious Thing	**Hard**	queued_q
H	Induction	**Hard**	g3gogogo
I	Quartet	**Challenging**	bnb2011
J	Cherry Blossom Ending	**Challenging**	leo020630

A. Table Tennis

`implementation`

Difficulty intention – **Easy**

- ✓ Statistics:
  - Junior: Submission - 22 times, AC - 17 times, AC rates - 77.273%
  - Senior: Submission - 49 times, AC - 29 times, AC rates - 59.184%
- ✓ First Solve:
  - Junior: **Beishekeev Daniyar**[UNIST], 2 min.
  - Senior: **김경민**[POSTECH], 2 min.
- ✓ Problem maker: Dongkyu Han[queued_q]

A. Table Tennis

- We can simulate the scores as given in the problem.
- Define variables X and Y.
- Using a loop, check N rounds and add a point to either X or Y. Using a conditional statement, if it becomes abs$(X - Y) \geq 2$, escape the loop using `break` statement.
- Finally, print the scores in the format `X:Y`.

B. UDPC party

`math`

Difficulty intention – **Easy**

- ✓ Statistics:
  - Junior: Submission - 103 times, AC - 13 times, AC rates - 12.621%
  - Senior: Submission - 120 times, AC - 28 times, AC rates - 23.333%
- ✓ First Solve:
  - Junior: **Amirbek Abdirasulov**[UNIST], 10 min.
  - Senior: **Mansur Mukimbekov**[UNIST], 8 min.
- ✓ Problem maker: Jeehoon Noh[g3gogogo]

B. UDPC party

- ✓ In this problem, the alphabet U and C, D and P are interchangeable, so we can treat U and C, D and P are the same character.
- ✓ Therefore, it satisfies the following two properties.
  - Both Dalgoo and Ponix may or may not be crowned. In other words, print both D and P or none.
  - Since the string that is the participants' vote is not empty, if Dalgoo and Ponix cannot be crowned, Yunee must be crowned. That is, the output is never C.
- ✓ Now, we can find out who can be crowned the cutest mascot through UC and DP - the number of U and C, the number of D and P in the participants' vote.

B. UDPC party

- For a mascot to be crowned, that mascot must receive strictly more votes than each of the other two mascots.
- Therefore, each mascot has to satisfy the following conditions to be crowned.
  - Yunee: $(DP + 1)/2 < UC$
  - Dalgoo, Ponix: $0 < DP$
- The solution is to print the alphabets of the mascots that satisfy the above conditions in order.

C. Card Flip

`constructive`

Difficulty intention – **Medium**

- ✓ Statistics:
  - Junior: Submission - 37 times, AC - 16 times, AC rates - 43.243%
  - Senior: Submission - 47 times, AC - 25 times, AC rates - 53.191%
- ✓ First Solve:
  - Junior: **Beishekeev Daniyar**[UNIST], 22 min.
  - Senior: **Mansur Mukimbekov**[UNIST], 16 min.
- ✓ Problem maker: Beomsu Kim[`qjatn0120baek`]

C. Card Flip

- ✓ First, you cannot flip over any cards after flipping over the card with the number N
- ✓ Therefore, you must flip the card with the number N last.
- ✓ Now you need to arrange the rest of the cards appropriately.

C. Card Flip

- ✓ First of all, the card with the number $N-1$ should be placed at the very end. Otherwise, you won't be able to flip over any other cards after flipping over the card with the number $N-1$.
- ✓ Now that you have flipped over the card with the number $N-1$, there are $N-1$ cards left to flip over. However, if you don't flip over the card with the number $N-2$ at this point, you won't be able to flip it over later. Therefore, after flipping over the card with the number $N-1$, you must flip over the card with the number $N-2$.

- ✓ If you continue this process, two types of arrangements are possible as shown below.
  - $[\ldots, 6, 4, 2, N, 1, 3, 5, \ldots]$
  - $[\ldots, 5, 3, 1, N, 2, 4, 6, \ldots]$

  In this case, the cards should be flipped in the order of $N-1, N-2, \ldots, 2, 1$ and N.

D. Coin Puzzle

`brute_force`

Difficulty intention – **Medium**

- Statistics:
  - Junior: Submission - 15 times, AC - 10 times, AC rates - 66.667%
  - Senior: Submission - 35 times, AC - 19 times, AC rates - 54.286%
- First Solve:
  - Junior: **Beishekeev Daniyar**[UNIST], 29 min.
  - Senior: **Mansur Mukimbekov**[UNIST], 24 min.
- Problem maker: Dongkyu Han[queued_q]

D. Coin Puzzle

- We need to find out where we should make the target shape.
- Suppose we will make the target shape at (a, b)-translated position. The number of coins to move is the number of coins in the current shape that does not belong to the translated target shape.
- Change each of a and b from -10 to 10, and then count the number of coins to move. We can choose (a, b) such that the number of coins to move becomes the smallest.
- We can implement it using a quadruple nested loop.
- Be careful about handling index out of range.

E. Hyundai Mobis Joining Project

`greedy, sorting`

Difficulty intention – **Medium**

- ✓ Statistics:
  - Junior: Submission - 16 times, AC - 9 times, AC rates - 56.250%
  - Senior: Submission - 41 times, AC - 24 times, AC rates - 58.537%
- ✓ First Solve:
  - Junior: **Beishekeev Daniyar**[UNIST], 34 min.
  - Senior: **오상규**[DGIST], 22 min.
- ✓ Problem maker: Dohun Kim[99asdfg]

E. Hyundai Mobis Joining Project

- ✓ We need to find the maximum sum of two types of abilities when selecting K lectures out of N lectures.
- ✓ Of course, it is impossible to solve the problem by finding all possible cases since the number of cases when selecting K lectures out of N lectures is ${}_NC_K$.

E. Hyundai Mobis Joining Project

- ✓ However, there are only three cases when choosing two abilities out of three to strengthen.
- ✓ Furthermore, once we have selected the two abilities to enhance, we can simply find the maximum sum of the two abilities by selecting K lectures with the maximum sum of the corresponding two abilities.
- ✓ Therefore, by finding the maximum sum for each of the three cases of choosing abilities to enhance, we can solve the problem within a time complexity of $\mathcal{O}(N \log N)$.

F. How-many-burgers can you eat at once?

`greedy`, `sorting`

Difficulty intention – **Hard**

- ✓ Statistics:
  - Junior: Submission - 17 times, AC - 4 times, AC rates - 23.529%
  - Senior: Submission - 58 times, AC - 14 times, AC rates - 24.138%
- ✓ First Solve:
  - Junior: **Amirbek Abdirasulov**[UNIST], 72 min.
  - Senior: **Mansur Mukimbekov**[UNIST], 38 min.
- ✓ Problem maker: Jeehoon Noh[g3gogogo]

F. How-many-burgers can you eat at once?

- ✓ The more coke-effects are stacked and the greater the mass of the hamburger eaten, the less the mass increased by eating the hamburger. That is, the mass of food in his stomach increases less.
- ✓ Therefore, when the coke-effect is stacked a lot, he has to eat a hamburger with a large mass to minimize the mass of food in his stomach, and we can use this to find the minimum value of the capacity of his stomach.
- ✓ Let $Count$ be a sorted array containing the number of coke-effects stacked when eating $1st, 2nd, ..., Nth$ hamburger, and $Mass$ be a sorted array containing the mass of hamburgers.

F. How-many-burgers can you eat at once?

- ✓ Since the order in which the hamburgers are eaten does not matter, we can calculate the minimum value of the capacity of his stomach with the formula below.

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

- ✓ At this time, the criteria for sorting the two arrays must be the same in ascending or descending order.
- ✓ The time complexity becomes $\mathcal{O}(KL)$ if you add the number of coke-effects stacked for each coke, so the time limit exceeded can occur.

F. How-many-burgers can you eat at once?

- ✓ Accordingly, we should reduce the number of computing operations by using some methods such as Prefix sum.
- ✓ For example, sort the timing of drinking cokes and increase the index i from 1 to N, increase CNT, which is the number of coke-effects stacked at the current index, by the number of times he drinks coke at the current index,
  save $i + L$, which is the index that this coke-effect ends and decrease CNT by the number of times the coke-effect ends at the current index,
  do $Count[i] = CNT$, we can solve this problem with $\mathcal{O}(K + L)$.

G. Yunee Stole the Precious Thing

`graph_traversal`
Difficulty intention – **Hard**

- ✓ Statistics:
  - Junior: Submission - 9 times, AC - 6 times, AC rates - 66.667%
  - Senior: Submission - 37 times, AC - 13 times, AC rates - 35.135%
- ✓ First Solve:
  - Junior: **정윤제**[POSTECH], 43 min.
  - Senior: **Mansur Mukimbekov**[UNIST], 46 min.
- ✓ Problem maker: Dongkyu Han[queued_q]

G. Yunee Stole the Precious Thing

- For a leaf node, if (the distance from Yunee) < (the distance from the police), Yunee can run directly towards this leaf node and escape there.
- Run DFS or BFS from each of Yunee, Dalgoo, and Ponix to calculate the distances to each leaf node d_a, d_b, d_c.
- If there exists a leaf node i such that $d_a(i) < d_b(i)$ and $d_a(i) < d_c(i)$, Yunee can escape. Otherwise, Yunee cannot escape.
- The time complexity is $\mathcal{O}(N)$.

H. Induction

DP

Difficulty intention – **Hard**

- ✓ Statistics:
  - Junior: Submission - 19 times, AC - 4 times, AC rates - 21.053%
  - Senior: Submission - 24 times, AC - 8 times, AC rates - 33.333%
- ✓ First Solve:
  - Junior: **Adil Sultanov**[UNIST], 88 min.
  - Senior: **Mansur Mukimbekov**[UNIST], 67 min.
- ✓ Problem maker: Jeehoon Noh[g3gogogo]

H. Induction

- Since the induction has three stoves, there are three ways to cook each dish.
  - 1) Set the first stove to the required temperature and cook the dish.
  - 2) Set the second stove to the required temperature and cook the dish.
  - 3) Set the third stove to the required temperature and cook the dish.
- We can set up an array DP indexed by the number of dishes cooked and the temperature of the three stoves.
- $DP[i][t_a][t_b][t_c]$: When the number of dishes cooked so far is i and the temperatures of the current three stoves are t_a, t_b, t_c in order, the minimum number of times he needs to touch the buttons so far.

H. Induction

- ✓ Then, when the temperature required to cook the $i+1_{th}$ dish is t_{i+1}, we can also set up the recurrence relation as below.
  - 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. Induction

- ✓ Since we want to find the minimum number of times he needs to touch the buttons, we must initialize the array DP with a value of at least $25\,000$.
- ✓ According to the fourth rule of the description, the temperature cycles from 0 to 9, so the number of times he touches the buttons to cook a dish does not exceed 5. $(5 \times 5\,000 = 25\,000)$
- ✓ Insert 0 to $DP[0][0][0][0]$, which is the initial state, and proceed according to the above recurrence relation from $DP[0][0][0][0]$ to $DP[N][9][9][9]$ for indexes other than the initial value.
- ✓ Then, the minimum value in the range from $DP[N][0][0][0]$ to $DP[N][9][9][9]$ is the minimum number of times he needs to touch the buttons to cook all the dishes in the given order, and we can solve the problem with $\mathcal{O}\left(1\,000N\right)$.

I. Quartet

graphs, ad hoc

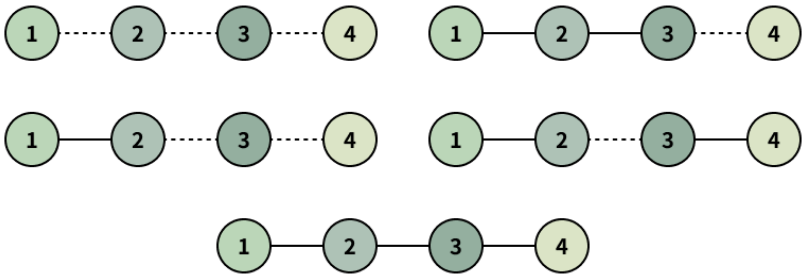
Difficulty intention – **Challenging**

- ✓ Statistics:
  - Junior: Submission - 19 times, AC - 0 times, AC rates - 0.000%
  - Senior: Submission - 32 times, AC - 4 times, AC rates - 12.500%
- ✓ First Solve:
  - Junior: -
  - Senior: **Mansur Mukimbekov**[UNIST], 117 min.
- ✓ Problem maker: Shinwook Park[bnb2011]

I. Quartet

- ✓ Let’s select four students in the order in which the seats are adjacent.
- ✓ Construct a graph with the student as the vertex and the synergy between the two students as the edge. The edges are weighted by the score of harmony that can be obtained when the synergy is satisfied.

- ✓ There are five cases where you can get a harmony score from an adjacent seating arrangement.



- ✓ The above four cases can be considered by selecting up to two synergies with the greatest score of harmony.
- ✓ There are two cases that need to be discussed in the problem.
  - Choose up to two edges in order of the greatest weights
  - Find the path with the largest sum of the weights of the edges among the simple paths consisting of three edges instead of a cycle

- ✓ The first case is simple. You can get the answer by sorting weights in ascending order or searching them twice. Time complexity is $\mathcal{O}(M)$.
- ✓ The second case is a little bit difficult. If a given graph is dense, the number of paths through three edges can be large, approximately $\mathcal{O}\left(N^3\right)$.

- Let’s fix the middle of the three edges.
- Once you’ve fixed the edges, you can construct a path by choosing one edge from each end.
- You don’t need to check all the edges connected to each vertex. It is sufficient to check only the few edges with the most weight.
- The constructed path should be simple. So you need to check the path does not form a cycle and does not use the same edge twice. Therefore, you only need to check each of the three edges in the order in which the weights are greatest.

- ✓ Finding the three most weighted edges per vertex can be done in $\mathcal{O}(N+M)$. And selecting both edges for holding the middle edges is also possible in $\mathcal{O}(1)$. per edge.
- ✓ Overall complexity is $\mathcal{O}(N+M)$.

J. Cherry Blossom Ending

`two_pointer`, `data_structure`
Difficulty intention – **Challenging**

- ✓ Statistics:
  - Junior: Submission - 0 times, AC - 0 times, AC rates - -%
  - Senior: Submission - 19 times, AC - 2 times, AC rates - 10.526%
- ✓ First Solve:
  - Junior: -
  - Senior: **Almas Abilkhanov**[UNIST], 162 min.
- ✓ Problem maker: Leo Jang[leo020630]

J. Cherry Blossom Ending

- ✓ Let's fix starting tree of successive blooming trees, and call this x.
- ✓ If you add a tree in the direction of increasing the number from the tree x, the blooming period of the trees overlaps with a certain tree, but no longer overlaps after that tree. Let's say this tree is D_x.
- ✓ And, if $i < j$, then $D_i < D_j$.

J. Cherry Blossom Ending

- So, we can get all D_x by two pointer.
- To determine if the blooming periods overlap, insert the start and end dates of the tree in the range you are looking at in a data structure such as a set and then check if $max(S_i) \leq min(E_i)$.
- You can also solve the problem by using parametric search and sliding window, or by using a segment tree.