B03902062 資工二 董文捷

Problem 1

1.

Fishy will be sad if f is larger than the maximum possible fish on the stand : **$f > s$** or t_i is so early that he is still not able to clean that much fish : **$f > (t_i - 1)$**

Fishy will be sad if **$f > \min(s, t_1 - 1)$**

2.

If for t_j ($j < i$) there is 1 fish on the stand and for t_i there is no fish on the stand, then Fishy should clean fish on $t_i - 1$ and that is an optimal solution since the fish will stay longer after t_i than any other solution.

So we can use a for loop to check if t_1, t_2, … still need fish. If yes, clean fish on $t_i - 1$, and that fish will stay on the stand on t_i, $t_i + 1$, … , $t_i + s - 1$, fish num of those time should increase by 1.

3.

If for t_j ($j < i$) there is f fish on the stand and for t_i still need n fish, then Fishy should clean fish on $t_i - 1$, $t_i - 2$, … , $t_i - n$ and that is an optimal solution since those fish will stay longer after t_i than any other solution.

So we can use a for loop to check if t_1, t_2, … still need fish. If t_i still need n fish, clean fish on $t_i - 1$, $t_i - 2$, … , $t_i - n$, and those fish will stay on the stand on (t_i, $t_i + 1$, … , $t_i + s - 1$), ($t_i - 1$, $t_i - 2$, … , $t_i + s - 2$) …

fish num of those time should increase correspondly.

Problem 2

1.

Equipment 1 is bought at time p_1 and will therefore increase influence b_1 during $p_1 \leqq t \leqq T$, equipment 2 is bought at time $p_1 + p_2$ and will therefore increase influence b_2 during $p_1 + p_2 \leqq t \leqq T$, ……

Average influence =

$$\frac{b_1 * (T - p_1 + 1) + b_2 * (T - p_1 - p_2 + 1) + \cdots + b_N * 1}{T}$$

2.

Before :

$$\frac{b_1 * (T - p_1 + 1) + \cdots + b_i * (T - p_1 - \cdots - p_i + 1) + b_{i+1} * (T - p_1 - \cdots - p_i - p_{i+1} + 1) + \cdots + b_N * 1}{T}$$

After :

$$\frac{\begin{gathered}b_1 * (T - p_1 + 1) + \cdots + b_{i+1} * (T - p_1 - \cdots - p_{i+1} + 1) \\ +b_i * (T - p_1 - \cdots - p_{i+1} - p_i + 1) + \cdots + b_N * 1\end{gathered}}{T}$$

Change = After – Before

=

$$\frac{b_{i+1} * \ p_i - \ b_i * \ p_{i+1}}{T}$$

3.

Assume there is an optimal solution A', we can prove A is also an optimal solution by swapping elements of A to get A'. At first, A is formed in the order A_1 A_2 …A_N. Suppose A' is formed in the order of A'_1 A'_2 …A'_N, we can swap A_j = A'_1 to the first position. By the condition, when swapping adjacent A_j and A_i with j > i, the new $f(A_{swap})$ is always smaller than or equal to f(A). So we can get an $A^{(1)}$ with A'_1 being the first element by swapping A_j with A_{j-1}, A_{j-2} … A_1, and $f(A^{(1)}) \leqq f(A)$ because the index j is larger than j – 1, j – 2, …, 1. After swapping A_j = A'_1 to the first position, other elements' index is still sorted, so we can repeat the process to get an $A^{(2)}$ with A'_2 being the second element, and $f(A^{(2)}) \leqq f(A)$. Finally, we can get A' by swapping elements of A, and $f(A') \leqq f(A)$ because each swap meet the condition of adjacent element with larger index on the right, which prove A is also an optimal solution.

4.

We can find an optimal solution A by 3, which means there does not exist any swap to become larger. By 2, to meet the condition, $\frac{b_j * p_i - b_i * p_j}{T} \leq 0$ for all j > i, which can be represent as $b_j / p_j \leqq b_i / p_i$ for all j > i. Therefore, we can calculate b / p (stored in double) for each equipment, then sort them in descending order according to that value, the final order after sort is the buying order which maximizes the average influence.

Time complexity :

Calculate b / p : O(n)

Sort : O(n log n)

Total : O(n log n)

Proof :

No such swap of A_i A_j in S is becoming larger because $b_j / p_j \leqq b_i / p_i$ for all j > i, $\frac{b_j * p_i - b_i * p_j}{T} \leq 0$ for all j > i, by 2, the change of average influence is negative or zero for all j > i, which means there does not exist any swap of A_i A_j in S to become larger for j > i. By 3, a sequence satisfying such condition is an optimal solution.

5.

We can also use $b_j * p_i - b_i * p_j$ (stored in long long) as a compare condition in sort function, if the result $\leqq 0$, means $b_j / p_j \leqq b_i / p_i$, so j should be bought later than i.

Problem 3

1.

If one choose the largest card on table in i-th pick, he can always get a card larger than or equal to $a_{[i]1}$. Since there are i cards larger than or equal to $a_{[i]1}$, and before i-th pick, only i – 1 cards are picked, at least one of those i cards will still on the table.

So Paul can use the strategy "Choose the largest card on table" to get a score at least $a_{[1]1} + a_{[3]1} + \ldots = \sum_{j\ is\ odd} a_{[j]1}$

John can use the same strategy to get a score at least $a_{[2]1} + a_{[4]1} + \ldots =$

$\sum_{j\ is\ even} a_{[j]1}$

2.

Paul has a strategy to get a score of $\sum_{i=1}^{N}\sum_{j=1}^{n_i/2} a_{ij}$. If last time John picks his j-th card of pile i, and Paul only picks j – 1 cards of pile i, then Paul should also picks his j-th card of pile i, too. In other situation, Paul can randomly choose one pile to pick (Including his first pick). The strategy ensure Paul to get top half of all piles, having a score of exactly

$\sum_{i=1}^{N}\sum_{j=1}^{n_i/2} a_{ij}$.

In the similar way, John has a strategy to get a score of $\sum_{i=1}^{N}\sum_{j=\frac{n_i}{2}+1}^{n_i} a_{ij}$.

If last time Paul picks his j-th card of pile i, John should also picks his j-th card of pile i, too. The strategy ensure John to get bottom half of all piles, having a score of exactly $\sum_{i=1}^{N}\sum_{j=\frac{n_i}{2}+1}^{n_i} a_{ij}$.

3.

This problem is quite similar to last problem. However, since n_i is odd, Paul and John can't divide each pile equally, let's call the card $a_{i(n_i+1)/2}$

“middle card”.
Paul has a strategy to get a score at least

$\sum_{i=1}^{N} \sum_{j=1}^{(n_i-1)/2} a_{ij} + \sum_{k \text{ is odd}} a_{[k](n_i+1)/2}$. If last time John picks his j-th card of pile i and it is not a middle card, and Paul only picks j – 1 cards of pile i, then Paul should also picks his j-th card of pile i, too. In other situation, Paul should pick the pile with largest middle card on the table.

If Paul picks a pile of odd card number (else branch), he is guaranteed to get the middle card of that pile because it indicates that pile has the largest middle card on table, whenever John pick the same pile, the strategy falls into else branch again, Paul will go on to pick card of that pile, until getting the middle card of that pile. Paul and John will take turns taking the middle card because after Paul gets one middle card, he will then follows John’s pick until John picks a middle card and Paul can pick the pile with largest middle card again. Therefore, Paul is ensured to get the top half of all piles = $\sum_{i=1}^{N} \sum_{j=1}^{(n_i-1)/2} a_{ij}$. He can also get the first, third, fifth middle card… By the strategy of pick the pile with largest middle card, middle cards’ score are at least $\sum_{k \text{ is odd}} a_{[k](n_i+1)/2}$

Sum up at least $\sum_{i=1}^{N} \sum_{j=1}^{(n_i-1)/2} a_{ij} + \sum_{k \text{ is odd}} a_{[k](n_i+1)/2}$

John can use the same strategy as above to get score at least

$\sum_{i=1}^{N} \sum_{j=(n_i+3)/2}^{n_i} a_{ij} + \sum_{k \text{ is even}} a_{[k](n_i+1)/2}$

4.
Combine 2 and 3, Paul can get a score S_Paul at least

$(\sum_{i=1}^{N\_even} \sum_{j=1}^{n_i/2} a_{ij})$ + $(\sum_{i=1}^{N\_odd} \sum_{j=1}^{(n_i-1)/2} a_{ij} + \sum_{k \text{ is odd}} a_{[k](n_i+1)/2})$

(result_1)
(N_even is the number of piles having even cards, and N_odd is the number of piles having odd cards)
Strategy : Use strategy of 3 to take piles of odd cards and only use strategy of 2 to take piles of even cards when John takes piles of even cards.
Following John’s pick on piles of even cards can ensure the result of 2, and pick first on piles of odd cards can ensure the result of 3.

John can get a score S_John at least by the same strategy as above

$(\sum_{i=1}^{N\_even} \sum_{j=\frac{n_i}{2}+1}^{n_i} a_{ij}) + (\sum_{i=1}^{N\_odd} \sum_{j=(n_i+3)/2}^{n_i} a_{ij} + \sum_{k \text{ is even}} a_{[k](n_i+1)/2})$

(result_2)

S_Paul $\geqq$ result_1

S_John $\geqq$ result_2

S_Paul + S_John = sum of all cards = result_1 + result_2

$\rightarrow$ S_Paul = result_1 S_John = result_2

If they all play optimally, Paul will get result_1 score and John will get result_2 score