NTUCSIE ADA 2015

Homework 3

B03902027 王冠鈞

Problem 1

(0) References:

 (i) Shared opinion with b03902028

(1) When $f > s \text{ or } f > t_1, Fishy\ will\ be\ really\ sad\ no\ matter\ how\ hard\ he\ struggles$

Description: For the first $f > s$, it means that the remain time of the fish are too short so that Fishy can't accumulate fish on the stand until it reaches the required value.

As for the second $f > t_1$, it means that there exists some time at of before $t = f$ that at that time it requires f fish on the stand, but it's too early that even if Fishy keeps cleaning the fish at every second, he still cannot catch the deadline.

(2) Description: Assume it's at t second now. Then, check whether the required fish is f = 1 or not at t + 1 second. If it isn't, continue; if it is, check the fishes on stand at t + 1 second. If it's not 0, continue; if it is, go clean a fish at the current time (time t) , increment the next s seconds' fishes on stand by 1, and then move to time t + 1.

Pseudocode:

```
Algorithm(Array of Table table, s)
for t = 1 to n
 if table[t + 1].required = 1 and table[t + 1].fish = 0
    table[t].clean_fish = true
    increment the next s seconds by 1
```

Prove correctness:

 (i) greedy choice property

 Let t_g be the first time to clean the fish, which is at $t_1 - 1$. If OPT contains t_g, then done. If not, there must be another t_h before t_g. Replace t_h into t_g, and it become another solution OPT', which is no worse than OPT.

 (ii) optimal substructure

 Let $m[t_i]$ be the optimal solution of t_i to t_n. Then the entire solution will be $m[t_1]$, and we know that according to the algorithm, Fishy will clean a fish at time $t_1 - 1$. Thus, the solution can be written as:

$$m[t_1] = \{t_1 - 1, m[t_k]\}, \text{where } t_k\ is\ the\ first\ time\ after\ t_1 + s$$

(3) Description: For every time t, scan the next f seconds first. If there exists any time within t + 1 and t + f at which time the required fish is f, check the current fishes on stand at that time. If it's equal to f, continue scanning until it reaches t + f. If it's smaller, calculate the time difference between that time and t, and the difference between the required quantity and current quantity at that time. If the time difference is larger, keep scanning until it reaches t

+ f; if not, clean a fish at the current time, increment the next s seconds' fishes on stand by 1, and then move to time t + 1.

Pseudocode:

```
Algorithm(Array of Table table, s, f)
for t = 1 to n
 for i = t + 1 to t + f
     if table[i].required = f
         if table[i].fishes < table[i].required
             time_diff = i - t
             fish_diff = table[i].required - table[i].fishes
             if time_diff = fish_diff
                 table[t].clean_fish = true
                 increment the next s seconds by 1
                 continue
```

Prove correctness:

(i) greedy choice property

Let $t_{g[1]}, t_{g[2]} \dots t_{g[f]}$ be the first f times to clean the fish in order to reach f fish on the stand so that it can meet the requirement at time t_1. If some of those $t_{g[i]}$ aren't in the current OPT, we can exchange the current $t_{g[i]}$s with the f times which are the nearest to time t_1(which are the times found from the algorithm above), and such solution is also at least no worse than the current OPT, thus composing a new OPT'. We can still construct a new OPT' even if there are already some fish on stand. The difference is that the number of times to find will be less than f.

(ii) optimal substructure

Consider the fish cleaned at time $t_{g[1]}, t_{g[2]} \dots t_{g[i]}, 1 \le i \le f$ all cover $t_1, t_2 \dots t_j$ completely. Let $m[t_i]$ be the optimal solution of t_i to t_n. Then, the entire solution $m[t_1]$ will be:

$$m[t_1] = \{t_{g[1]}, t_{g[2]}, \dots, t_{g[f]}, m[t_{j+1}]\}$$

, then we can continue find the optimal solution of the sub-problem $m[t_{j+1}]$.

Or, we can construct another expression:

$$m'[t_1][t_{t[1]}][t_{t[2]}] \dots [t_{t[f]}]$$

, where $t_{t[i]}$ denotes the last second that i-th fish remains, and when the current time is over the last second, it will be reset to 0 and allow further modification. Thus, in this expression,

$$m[t_1] = m'[t_1][t_1 + s - f][t_1 + s - (f-1)] \dots [t_1 + s - 1]$$
$$= \{t_{g[1]}, t_{g[2]}, \dots, t_{g[f]}, m'[t_{j+1}][0] \dots [t_1 + s - 1]\}$$

Problem 2

(0) References:

(ii) Shared opinion with b03902028

(1) Average influence $= \frac{\sum_{1\le t\le T}\sum_{i\in E_t} b_i}{T}$

$$= \frac{(T-p_1)b_1 + (T-p_1-p_2)b_2 + \cdots + (T-p_1-p_2-\cdots-p_N)b_n}{T}$$

$$= \frac{T\cdot\sum_{i=1}^{N} b_i - [p_1(b_1+b_2+\cdots+b_N) + p_2(b_2+\cdots+b_N) + \cdots + p_N(b_N)]}{T} \cdots (1)$$

$$= \sum_{i=1}^{N} b_i - \frac{\sum_{i=1}^{N} p_i \sum_{j=i}^{N} b_j}{T}$$

(2) Extract the $\mathrm{p_i}(b_i + \cdots + b_N)$ and $\mathrm{p_{i+1}}(b_{i+1} + \cdots + b_N)$ terms of (1) and compare with the new $\mathrm{p_i}(\dots), p_{i+1}(\dots)$ terms below:

new average influence =

$$\sum_{i=1}^{N} b_i - \frac{p_1(\sum_{k=1}^{N} b_k) + \cdots + p_{i-1}(\sum_{k=i-1}^{N} b_k) + p_{i+1}(b_i + b_{i+1} + \cdots + b_N) + p_i(b_i + b_{i+2} + \cdots + b_N) + \cdots + p_N(b_N)}{T} \quad (2)$$

We now take $p_i(b_i + b_{i+2} + \cdots + b_N)$ and $p_{i+1}(b_i + b_{i+1} + \cdots + b_N)$ of (2) for comparison with $\mathrm{p_i}(b_i + \cdots + b_N)$ and $\mathrm{p_{i+1}}(b_{i+1} + \cdots + b_N)$ of (1), and find out:

$$p_i(b_i + b_{i+2} + \cdots + b_N) + p_{i+1}(b_i + b_{i+1} + \cdots + b_N)$$
$$= \mathrm{p_i}(b_i + \cdots + b_N) + \mathrm{p_{i+1}}(b_{i+1} + \cdots + b_N) + \mathrm{p_{i+1}}b_i - p_i b_{i+1}$$

Thus, new average influence

$$= \sum_{i=1}^{N} b_i - \frac{\sum_{i=1}^{N} p_i \sum_{j=i}^{N} b_j - p_i b_{i+1} + \mathrm{p_{i+1}} b_i}{T}$$

$$= \text{average influence} + \frac{p_i b_{i+1} - p_{i+1} b_i}{T}$$

and the change is:

$$\Delta(\text{average influence}) = \frac{p_i b_{i+1} - p_{i+1} b_i}{T}$$

(3) I want to know that given a sequence $\mathrm{A\{a_i\}}$, where $|\mathrm{A}| = \mathrm{N}$, for every $\mathrm{a_i}$ consisting of two integer values $\mathrm{p_i}, b_i$, then:

$\forall 1 \le \mathrm{i} \le \mathrm{N}-1,\ p_i b_{i+1} - p_{i+1} b_i \le 0 \iff \forall 1 \le \mathrm{i} < \mathrm{j} \le \mathrm{N},\ p_i b_j - p_j b_i \le 0 \iff \mathrm{A}$ is optimal

I first prove the left two-way arrow:

if($\Leftarrow$): trivial

only-if($\Rightarrow$): Use mathematical induction:

(i) base case: $j = i + 1$: trivial, that's in the statement above

(ii) inductive case: if $\forall i \le r < s \le k+1, \text{and } |r-s| \le k - i,\ p_i b_j - p_k b_j \le 0 \Rightarrow$ $p_i b_{k+1} - p_{k+1} b_i \le 0$

After lots of hand-written study and induction, I've got the difference of swapping i-th and k-th element (now consider the numerator and omit the denominator T first since it's always the same), which is:

$$\sum_{m=i}^{k} \sum_{n=m+1}^{k+1} p_m b_n - \sum_{m=i+1}^{k+1} \sum_{n=i}^{m-1} p_m b_n$$

since for all $|m-n| \le k - i, p_m b_n - p_n b_m < 0$ (3), due to the assumption of the inductive case, then we only need to test the case of terms that satisfy $|m-n| = k + 1 - I$, which contains only $p_i b_{k+1} - p_{k+1} b_i$. Thus the statement become simplified into proving that $p_i b_{k+1} - p_{k+1} b_i < 0$.

We can prove such statement by using the following inequalities which are extracted from (3) with specific m and n:

$$p_i b_k - p_k b_i \le 0 \ \cdots (4)$$
$$p_k b_{k+1} - p_{k+1} b_k \le 0 \cdots (5)$$

Now multiply something to (4) and (5):

$$p_{k+1} b_{k+1}(p_i b_k - p_k b_i) \le 0$$
$$p_i b_i (p_k b_{k+1} - p_{k+1} b_k) \le 0$$
$$\Rightarrow p_{k+1} b_{k+1}(p_i b_k - p_k b_i) + p_i b_i (p_k b_{k+1} - p_{k+1} b_k) \le 0$$
$$\Rightarrow (p_k b_i + p_{k+1} b_k)(p_i b_{k+1}) - (p_i b_k + p_k b_{k+1})(p_{k+1} b_i) \le 0$$
$$\Rightarrow \left(\frac{p_k b_i + p_{k+1} b_k}{p_i b_k + p_k b_{k+1}} - 1\right)(p_i b_{k+1}) + (p_i b_{k+1} - p_{k+1} b_i) \le 0$$

Note that the fraction $\frac{p_k b_i + p_{k+1} b_k}{p_i b_k + p_k b_{k+1}}$, observing with inequalities (4) and (5), and we can get that the fraction is larger than 1, so when it is subtracted by 1, it's still not less than 0. Thus:

$$p_i b_{k+1} - p_{k+1} b_i \le 0$$

, and we finished the proof.

Now, I'll prove the right two-way arrow:

only-if($\Rightarrow$): If the left statement is true, it implies that whichever two elements in sequence A we swap, the change of the average influence will always be negative. In other words, we will make the new average influence even lower if we swap any of the two elements. Thus, in this state, sequence A is optimal.

if($\Leftarrow$): Consider a $\sim Q \Rightarrow \sim P$ statement. If the left statement is false, then there exists a pair $(a_p, a_q), q > p$, such that when we swap them, the value: $p_p b_q - p_q b_p$ is positive, then the new average influence will increase.

Thus, all statements are proved.

(4) Use merge-sort. Although in a merge-sort algorithm, it may compare two elements which are not adjacent, but according to the result of the former sub-problem, it says that in a sorted sequence, if p is next to q, and the $g(p, q) = \mathrm{p_p b_q} - \mathrm{p_q b_p}$ value is negative, then for any $\mathrm{r} > \mathrm{q}, g(p, r)$ remains negative. Then it implies that the indices of the elements in the optimal state are absolute, not relative, so it is in fact isomorphic with an array of integers to sort by value. Thus, if merge-sort is correct for sorting integer arrays, it should be also correct for sorting the optimal buying sequence; the only difference between these two sequences is the comparison function. The latter one applies $g(p, q)$ as its comparison. The time complexity of the algorithm is $\mathrm{O}(\mathrm{n} \log n)$, since its essence is merge-sort.

(5) I didn't use any floating-point operation because the algorithm only calculates the $g(p, q)$ function, which is always in integers. However, to prevent overflow, the type that stores $g(p, q)$ can be modified into a 64-bit integer.

Problem 3

(0) References

(i) Shared opinion with b03902107

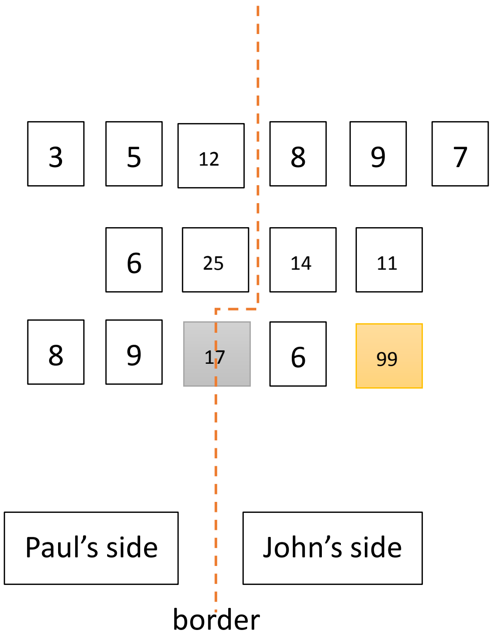
(1) Both strategies are to take the card of the largest number among the remaining cards.
If Paul applies the strategy, he will get at least $\sum_{j\ is\ odd} a_{[j]1}$. The minimum occurs when John applies the same strategy.
If John applies the strategy, he will get at least $\sum_{j\ is\ even} a_{[j]1}$. The minimum occurs when Paul applies the same strategy.

Prove Paul will get at least $\sum_{j\ is\ odd} a_{[j]1}$:
Since he always get the largest number of the remaining cards in every turn, in the first turn he will definitely get the largest, i.e. $\mathrm{a}_{[1]1}$. For the remaining turns, if John takes the largest ones, since Paul will be the person who can take a card in 3rd, 5th, 7th …(odd) turns, so he will take the 3rd, 5th, 7th … largest cards, which can be all written in $a_{[j]1}, j\ is\ odd$. If John doesn't, John will have the chance to get 2nd , 4th , 6th,… cards, which are larger than 3rd, 5th, 7th … ones, respectively. Thus, in this case, he will get larger than $\sum_{j\ is\ odd} a_{[j]1}$ in total.
The proof for John is basically the same.

(2) From this sub-problem, I'll introduce a way of thinking for this game, and let call it "*territory dividing*" for concrete comprehension, and the following strategies and optimal choices are based on this idea.
Consider a pile of cards (call it P) containing N cards inside, where N is even. Since it's a turn-based game, the number of cards each player can get has been determined as soon as the piles of cards are set. Furthermore, getting a good card from somewhere takes a player some turns to get, which should be regarded as *implicit cost* (a term in economics) of a player. In

other words, if the number of one of the first cards on John's side is far larger than any other cards in the entire piles (both players can see all cards in the beginning, as TA informed), Paul may covet that one (for example, the card "99" in the figure below), because if he could get it, it would become a great benefit for increasing the total score of his. However, while Paul must take all the other cards in that pile in order to get the very card, John can directly get that card immediately. Thus, not only getting such cards takes a player lots of turns, but it is also risky, since Paul cannot predict the next steps John will go. A player may end up losing even more scores if he merely focus on the globally-largest cards throughout the game. Thus, "taking the cards in the pile containing globally-largest card" isn't optimal, and I'll next implement the "*territory dividing*" idea and redefine the optimal choices during the game.

Let's go back to the initial part of the last paragraph. There are N cards in a pile P, and N is even. If for the next N turns, John and Paul get a card from pile P in every turn, then P will be empty, and Paul will get the top half, John getting the bottom half. Let's define a noun "*territory*" as the portion of an even pile of size N that a player can get if both player takes the cards in the pile only, and define the noun "*border*" to be the edge of the territory. In the case mentioned above, Paul's "territory" is the sequence of the top half of pile P, and John's is the bottom half. The border of pile P is between the N/2-th and (N/2)+1-st card from top. When a player takes a card from pile P, the border will move ahead for half unit, which will move onto a card, making the pile be odd. In the case of an odd pile Q, all the cards above the card in the midst (call it m) is the territory of Paul, below which card is the territory of player John. As for card m, it's not the territory of either Paul or John at the moment, and let's define it as an "*unoccupied territory*" (17 in the figure). However, if next turn is Paul, then Paul has an *"occupation advantage"*, which means Paul can take a card of that pile to move the border ahead for a half more unit, making the entire card be within his new territory.



As we know, a territory is defined as the cards in a pile that a player is guaranteed to get if two players keep taking cards in the same pile. Second, we know that the total number of cards a player can get has been determined in the very initial part. That is, if a player occupies a new territory which is at first belonged to the other player, he will lose a territory or his, or a chance to occupy an unoccupied territory, otherwise it will violate the rule of the above. Third, since a player is promised to get the cards in his territory by using appropriate defense (in order to push the territory back), and other than those cards, the cards which are most likely to get are the cards in unoccupied territories (which can be directly occupied in one turn) in odd piles, and then the first cards after crossing the borders in even piles (requiring two turns). Thus, we can consider the optimal choices to be exchanging the smaller territories into larger territories.

Let us redefine the optimal choices: select the card in the pile which requires a lowest *opportunity cost* (also a term in economics); that is, choose the pile whose "unoccupied territory" or the "first card after crossing the border" has the largest number. If there are more than one such piles, choose "unoccupied territory" for its higher priority, and if there are more than one such piles with a same classification, seek the next card to get, and choose the larger/largest pile (e.g. there are two piles with largest "first card after crossing the border", then seek the "second card after crossing the border", and choose the larger one. If they're still the same, keep seeking down until the two piles have different value on the i-th card.). Such strategy can be applied in sub-problems 2, 3 and 4.

Prove that Paul will get at least $\sum_{i=1}^{N}\sum_{j=1}^{n_i/2} a_{ij}$:

If John always take a card from the same pile right after Paul does, or applies the same strategy, Paul will get the scores of the sum of his original territory, that is: $\sum_{i=1}^{N}\sum_{j=1}^{n_i/2} a_{ij}$, the same as the value in the statement to prove.

If John takes a card from a pile with smaller "unoccupied territory" or the "first card after crossing the border", when there exists a larger unoccupied territory, Paul can take that, which will make his total score larger; when there exists a larger "first card after crossing the border", Paul can take one card from that pile, making that larger card become an unoccupied territory, and he can get it if John still doesn't take a card from that pile, and eventually make his total score larger. Even if John takes a card from the pile with the largest "first card of Paul's side after crossing the border", if there aren't any larger cards to covet, Paul can defend his card immediately since it's his advantage to get an unoccupied card, or if there are, Paul can take a card from that pile for a chance to exchange a larger card from his. Thus, whatever the opponent chooses, Paul can always get the result which is no worse than the minimum value, $\sum_{i=1}^{N}\sum_{j=1}^{n_i/2} a_{ij}$.

Prove that John will get at least $\sum_{i=1}^{N}\sum_{j={n_i}/_{2}+1}^{n_i} a_{ij}$:

Basically the same.

(3) Apply the same strategies as (2) mentioned.

Prove that Paul will get at least $\sum_{i=1}^{N}\sum_{j=1}^{\frac{n_i-1}{2}} a_{ij} + \sum_{k\ is\ odd} a_{[k]\frac{n_i+1}{2}}$:

If John applies the same strategy, we can imagine what will happen: in the initial state, there are N odd piles of cards, so they will definitely attempt to get the largest card among the unoccupied territories. After Paul occupies the largest first, in John's perspective, among the "unoccupied territories" and "first cards after crossing the borders", the largest one is still in the pile that Paul has chosen. Thus, they will end up cleaning up piles with largest unoccupied territory to the smallest, and for every pile, they are originally odd, so Paul and John will take turns to get the original unoccupied territories from the largest to the smallest, that is, Paul will get the 1st, 3th, 5th… (odd), and John will get the 2nd, 4th, 6th… (even), and the result of Paul's score will end up to be: $\sum_{i=1}^{N}\sum_{j=1}^{\frac{n_i-1}{2}} a_{ij} + \sum_{k\ is\ odd} a_{[k]\frac{(n_i+1)}{2}}$, which is the same value in the statement to prove.

If John doesn't apply it, just as the proof that sub-problem (2) has mentioned, Paul will always have a chance to exchange a territory with a larger value with his original one/unoccupied territory, which will definitely be no worse than the minimum value.

Prove that Paul will get at least $\sum_{i=1}^{N}\sum_{j={n_i+3}/_{2}}^{n_i} a_{ij} + \sum_{k\ is\ even} a_{[k]\frac{n_i+1}{2}}$:

Basically the same!

(4) If there are M even piles and N odd piles, they will get:

Paul: $\sum_{i=1}^{M}\sum_{j=1}^{n_i/2} a_{ij} + (\sum_{i=1}^{N}\sum_{j=1}^{\frac{n_i-1}{2}} a_{ij} + \sum_{k\ is\ odd} a_{[k]\frac{n_i+1}{2}})$

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

, which is the aggregated result of sub-problems (2) and (3).

Why? Because the strategy which is applied in (2) and (3) is considered optimal. The proofs in (2) and (3) implies that such strategy has a greedy-choice property. And also, it also has optimal substructures. Let $\mathrm{m}[\mathrm{E}_1[1,\mathrm{n}_{\mathrm{E}1}]] \dots [\mathrm{E}_\mathrm{M}[1,\mathrm{n}_{\mathrm{EM}}]][\mathrm{O}_1[1,\mathrm{n}_{\mathrm{O}1}]] \dots [\mathrm{O}_\mathrm{N}[1,\mathrm{n}_{\mathrm{ON}}]] = value\ of\ the\ optimal\ score\ that\ Paul\ gets\ while\ John\ also\ plays\ optimally$ $(E_i\ is\ an\ even\ pile, O_i\ is\ an\ odd\ pile\ in\ initial, and\ [a,b] are\ the\ numberings\ of\ the\ cards\ in\ the\ pile\ of\ initial\ state, n_K\ is\ the\ number\ of\ cards\ in\ pile\ K\ initially)$, then consider O_i has a largest "unoccupied territory"/ "first card after crossing the border" among all N+M tiles, and $\mathrm{O}_\mathrm{i}[1]$ is in the OPT of Paul, then after a cycle of turns, The scores of Paul will be:

$$\mathrm{O}_\mathrm{i}[1] + \mathrm{m}[\mathrm{E}_1[1,\mathrm{n}_{\mathrm{E}1}]] \dots [\mathrm{E}_\mathrm{M}[1,\mathrm{n}_{\mathrm{EM}}]][\mathrm{O}_1[1,\mathrm{n}_{\mathrm{O}1}]] \dots [\mathrm{O}_\mathrm{i}[2,\mathrm{n}_{\mathrm{Oi}}-1]] \dots [\mathrm{O}_\mathrm{N}[1,\mathrm{n}_{\mathrm{ON}}]]$$

So eventually they will get the score of the results of sub-problems (2) + (3) in the end.