Mini Comp 2 Editorial

Credits: Steven, Tony, Gagguy, LMH, Theo

HKOI BBQ

Setter: Steven

Prepared By: Steven

HKOI BBQ

IQ question:

3 chicken wings, 1 fork, how to bbq in minimal time?

A:1.5 * cooking time of 1 chicken wing

HKOI BBQ

N food, M forks, how to bbq in minimal time?

If we have blenders,

minimal time = sum(Ti) / (2M)

Without blenders,

minimal time = max(max(Ti), sum(Ti) / (2M))

HKOI BBQ

Yes, that's it.

But human make mistakes on stupid things.

- float
- int

Please, use double instead.

HKOI BBQ

The background story was fake,
but we do hope to BBQ with you one day!

Gyeolhap

Setter: Tony

Prepared By: Tony

Gyeolhap

No solution when N = 7, 9, 10 or 11

Use 12 bits to represent a piece

Why? It is easier to hard code them

0x111, 0x112, 0x114, 0x121, 0x122, 0x124

Gyeolhap

```
 0: 111 112 121 122 211 212 224 244 424
 1: 111 112 114 121 122 211 212 221 222
 2: 111 112 114 121 122 124 211 212 221
 3: 111 112 114 121 122 124 211 212 214
 4: 111 112 114 121 122 124 211 212 411
 5: 111 112 114 121 122 124 141 211 212
 6: 111 112 114 121 122 124 141 211 411
 7: 0 0 0 0 0 0 0 0 0
 8: 111 112 114 121 122 124 141 142 211
 9: 0 0 0 0 0 0 0 0 0
10: 0 0 0 0 0 0 0 0 0
11: 0 0 0 0 0 0 0 0 0
12: 111 112 114 121 122 124 141 142 144
```

Fix top left corner = 111

Exhaust all possible boards

- Recursion
- Since the number of items is fixed, we can alternatively write 9 layer of nested for-loop

Turns out that we can find the answer in very few tries

Image on the right shows the lexicographically smallest soln

Gyeolhap

Write a function to count the number of Haps

- Exhaust 012, 013, 014, …, 018, 023, …, 678
- For each group of 4 bits
- It has to be 1 or 2 or 4 or 7

If the number of Haps found = N, output

- Set 1 = char desired
- Set 2 and 4 = other two chars

The Migration

Setter: Gagguy

Prepared By: LMH

The Migration

Setter: Gagguy

Prepared By: LMH

The Migration

Two possible positions for each villager i (li, ri)

Minimize leftmost (L)- rightmost (R) coordinate

max(li, ri) - min(li, ri)

Exhaust all possible positions for every villager

Maintain L and R

O(2^n)

The Migration

Case 1 - L belongs to one of li

Assuming each li as L

Finding its minimum possible R in n - 1 others' (lj, rj)

When rj is smaller than li, assumption is wrong

Case 2 - L belongs to one of ri

Similar to Case 1

O(N^2)

The Migration

Sort all (l, r) by l

Case 1 - L belongs to one of li

Assuming each li as L starting from i = 1

Finding minimum possible R in O(1)

For 1 .. (i - 1) : The left bound is smaller or equal to li, therefore not possible to choose them as final positions. Otherwise, li would not be L

Maintain max r after each assumption. If max r is smaller than li, assumption is wrong again

For (i + 1) .. n : The left bound is bigger or equal to li, but only lN is possible to be R. Otherwise, neither lN nor rN are not possible for migration to that villager.

Maximum of these two values is R

The Migration

Case 2 - L belongs to one of ri

- Check only for smallest ri (Others are obviously impossible to be L)
- Finding its minimum possible R in n - 1 others' (l, r)
  - When rj is smaller than li, assumption is wrong

O(N log N)

Card Game

Setter: Theo

Prepared By: Theo, Tony

Card Game

What is Bob’s optimal strategy?

Intuitively Bob wants to deal as much damage as he can in limited round, an optimal sequence is to use up all the ice cards, and follow each ice card with a fire card.

((If there is not enough fire card we can insert some 0-damage fire cards.

Card Game

Obviously Bob will choose the cards (ice and fire) with highest damage first.

--> We can know that the number of rounds Bob need to defeat Alice. Let it be Y.

(If Bob cannot defeat Alice, Y = inf)

Card Game

Let Alice have X ice cards on hand (after shopping). Then, the number of fire cards Alice can deal is X + Y.

Subtask 3: t_i = 1, therefore X+Y is fixed (as she cannot buy ice cards from shop)

Card Game

Alice's best strategy is to output the (X+Y) fire cards with highest damage. If she can do so with her initial cards, then we don't need to buy any fire cards from shop.

What if she cannot do so?

Card Game

One best strategy to buy cards is to buy the strongest fire card, then if it's not enough, buy the second strongest, etc.

Let's simulate from the first bought card.

Now Alice has (X+Y+1) cards, but she can only deal (X+Y) of them. --> Remove the weakest among the cards.

Card Game

We can use a heap structure, as it supports:

--> Insert a card into the heap

--> Remove the weakest card from the heap

in O(lg P), where P is the size of heap, and P can never exceed (N+K)

Card Game

What if we still cannot defeat Bob with 1 fire card bought from shop? Continue buying!

As we can buy at most K cards from shop,
The complexity is O(N lg N + K lg (N + K))

Expected score: 10 :D

Card Game

Subtask 1 & 2: (N,M,K) <= 10^3

Now we have ice cards available in shop, therefore X+Y is no longer constant.

Let’s assume Alice has Z ice cards on hand, and she will buy Q ice cards from shop,

--> Q+Z = X

Card Game

If we fix Q, then X+Y will become constant.

How many ways can we select Q? at most 10^3

Also, intuitively Alice will always buy the strongest Q ice cards in shop.

For each Q, try the procedure described previously.

Complexity: (K(N lg N + K lg (N + K))

Card Game

Subtask 4: (N,M,K) <= 10^5

Let's assume f(Q) be the number of fire cards Alice need to buy if Alice bought Q ice cards from shop.

Easy fact: f(Q - 1) >= f(Q).

Card Game

Let's fix Q = (T, which T is number of ice cards in shop) first. I.e. initially we buy all the ice cards in shop.

We run the procedure described in Subtask 3.

What if we now fix new Q = T - 1?

as f(T - 1) >= f(T), we will keep buying new cards (but not discarding old cards!)

Card Game

--> for the case Q = T - 1, we can skip the first f(T) cards as they will always be bought.

We can keep the heap we used in f(T), with an exception of poping 1 card from the old heap (as we can put 1 fewer card!)

And then we continue buying cards from shop.

Card Game

After calculating f(T-1), we can continue to f (T-2), f(T-3)...f(0). Ans = min(f(x)+x) for each x from 0 to T.

Complexity?

Each card will be pushed and popped from card once. Complexity: O(N lg N + K lg (N + K))