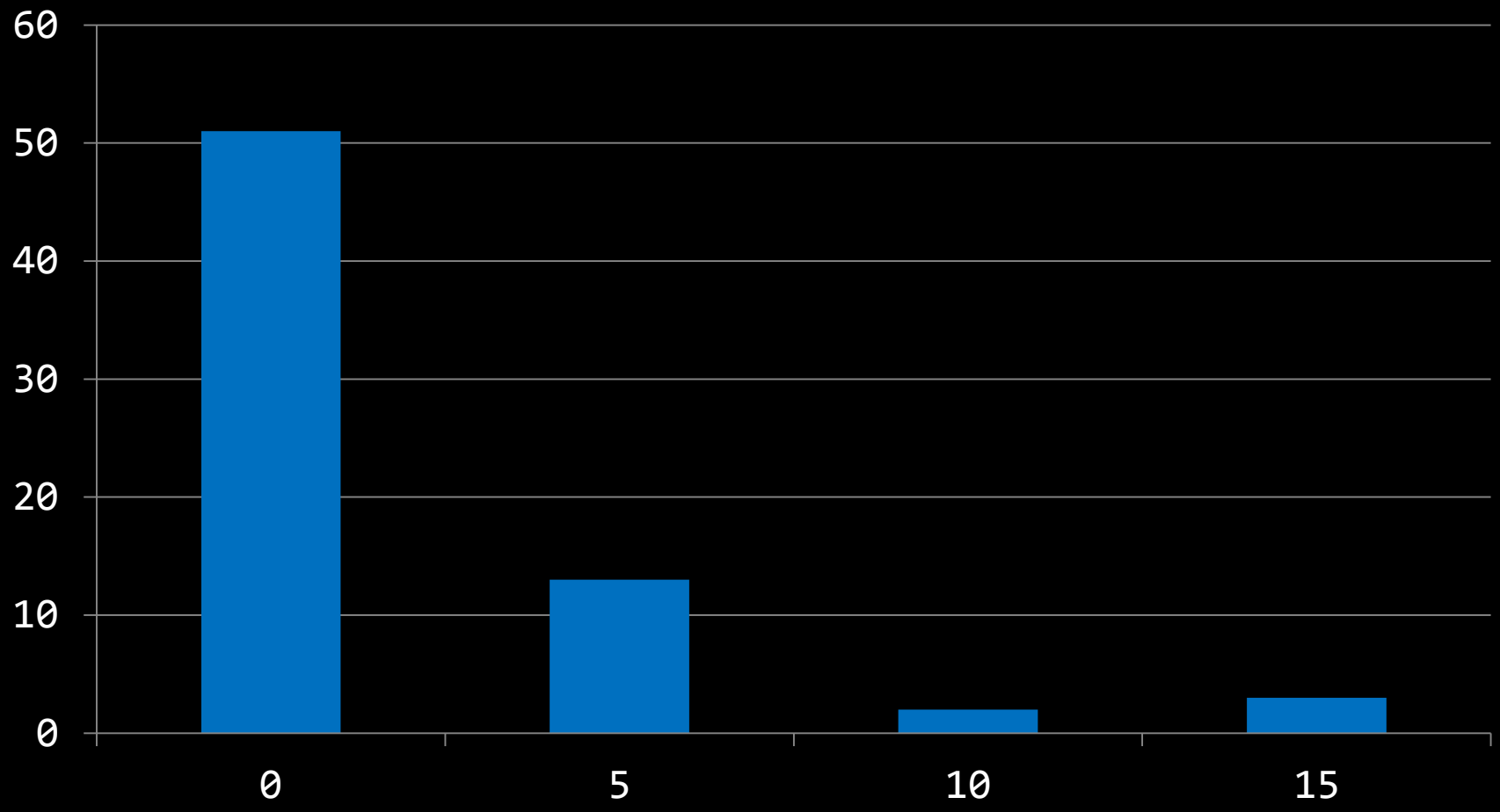
Infinite Coding

HKOI 2012 Senior Q1

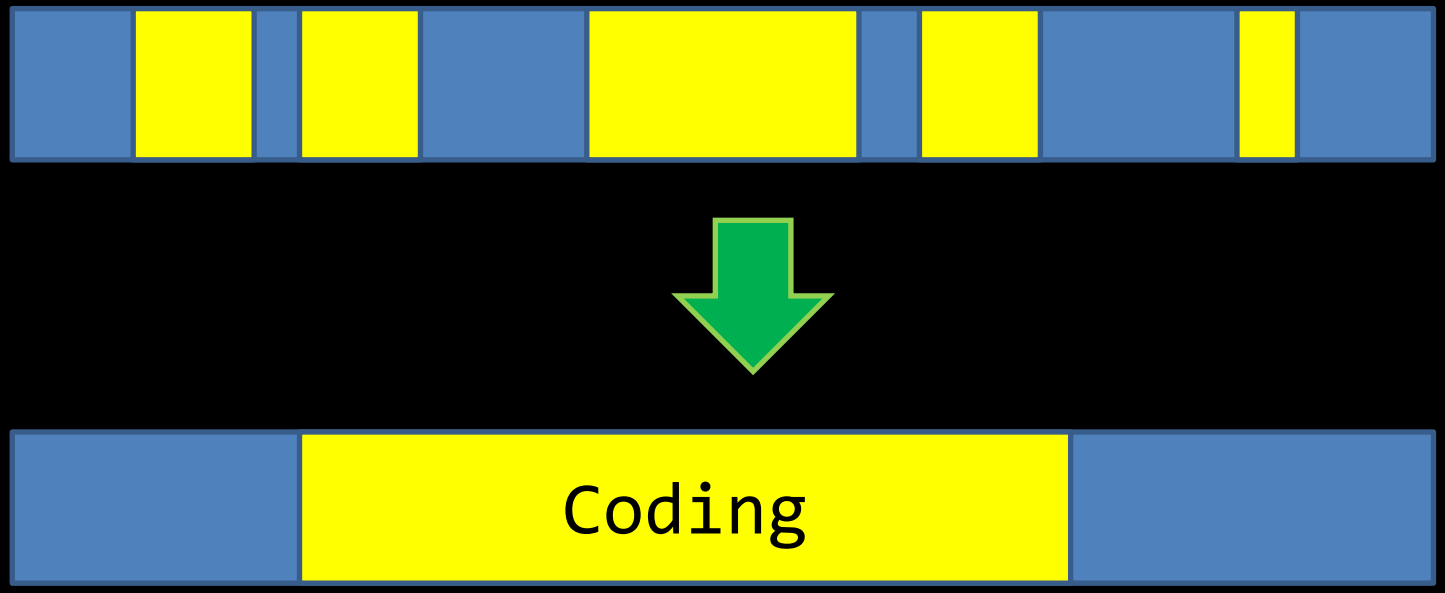

60
50
40
30
20
10
0
0
5
10
15

Problem

- N activities
- M of them are coding
- ‘Cut and paste’ operation with cost w[i]
- Minimum cost to group all coding

Solution

- First, lets assume that we **cannot** cross midnight



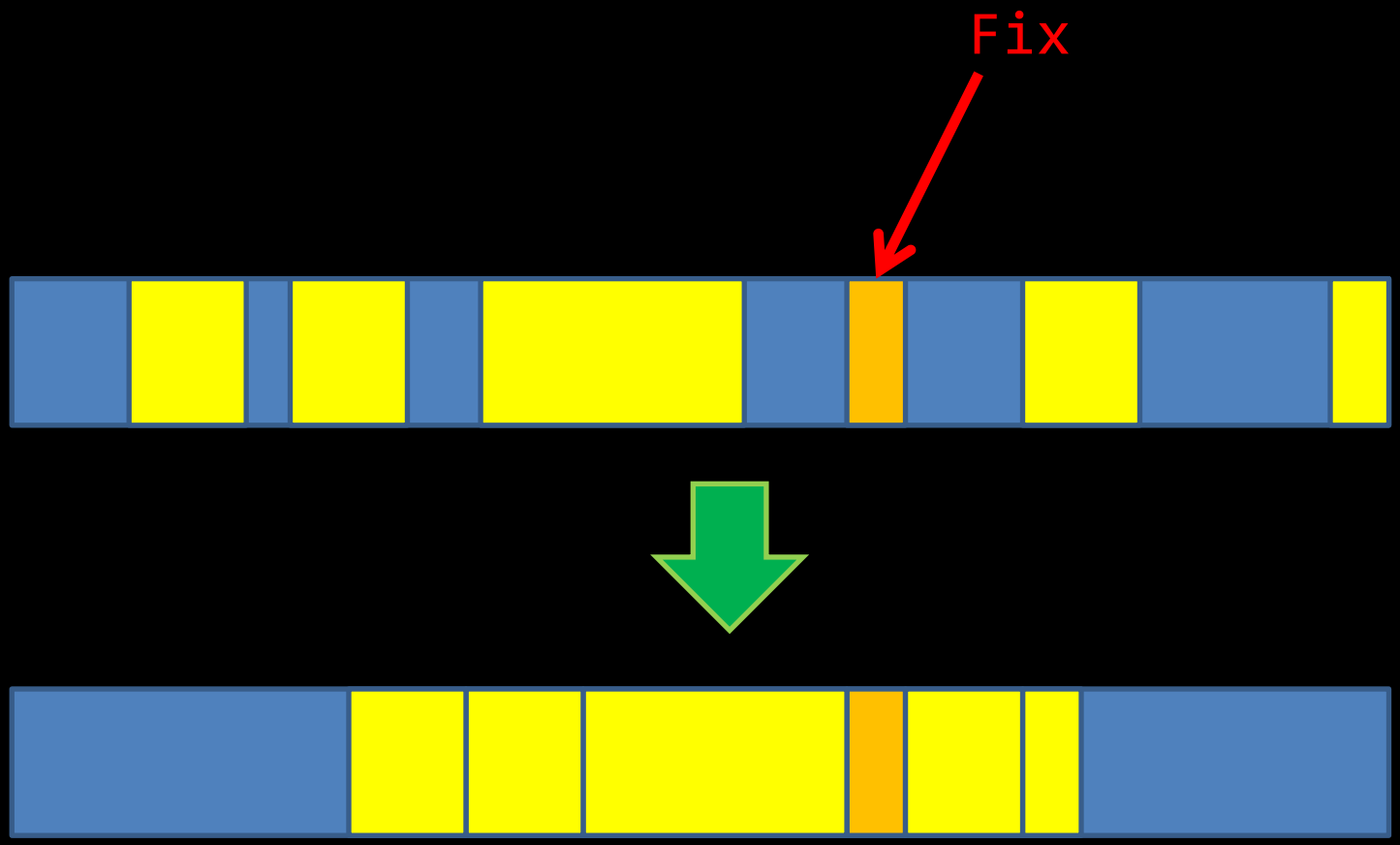
Observation

- Lemma: There exists a optimal solution that do not involves moving all coding sessions

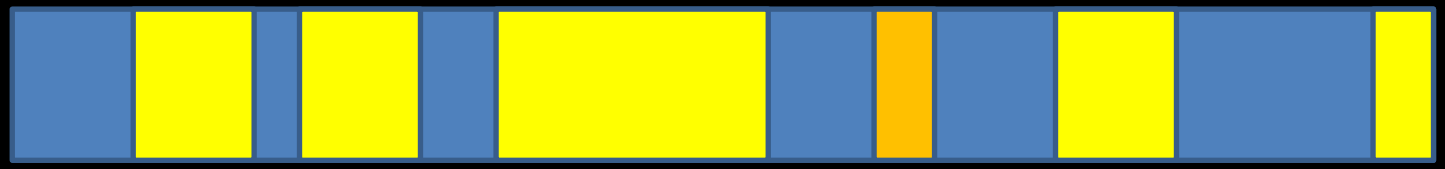
- Proof: We can always group all coding sessions by moving M-1 of them

Choose a coding session and fix it

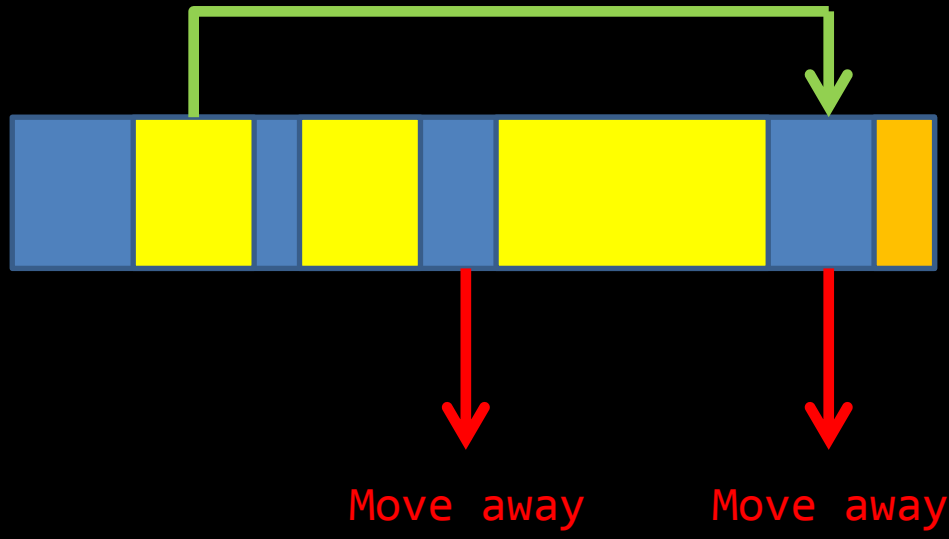
Solution



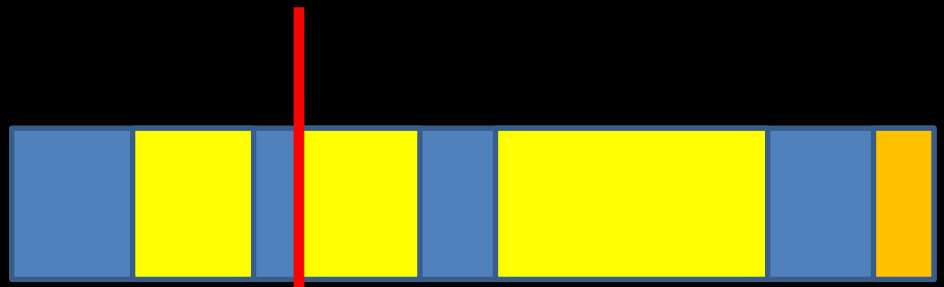
How do we group them



Focus on the left side..



How do we group them



Left of the line:

Move the coding sessions to the fix one

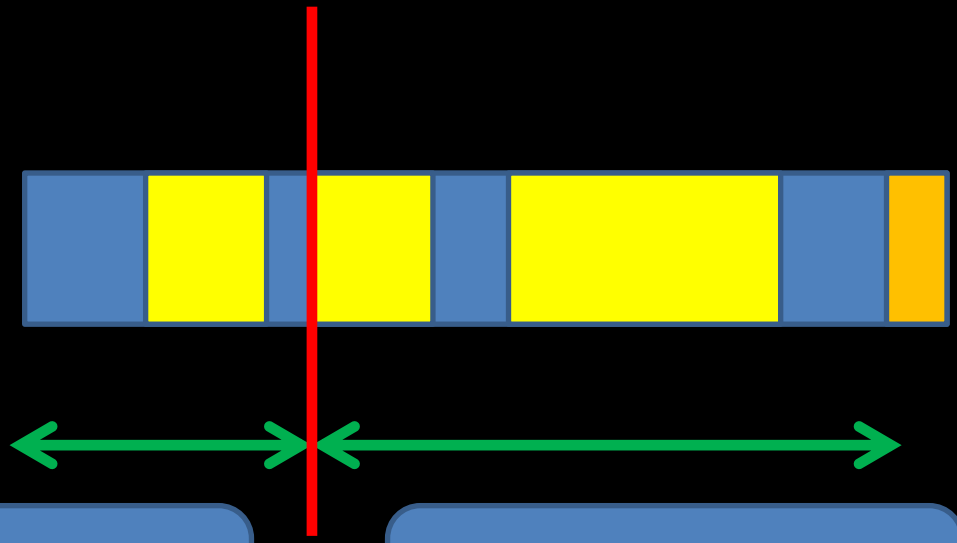
Right of the line:

Move the non-coding sessions away

How to find the optimal line?

How do we group them

Try all possible position



Sum up the cost of coding session

+

Sum up the cost of non-coding session

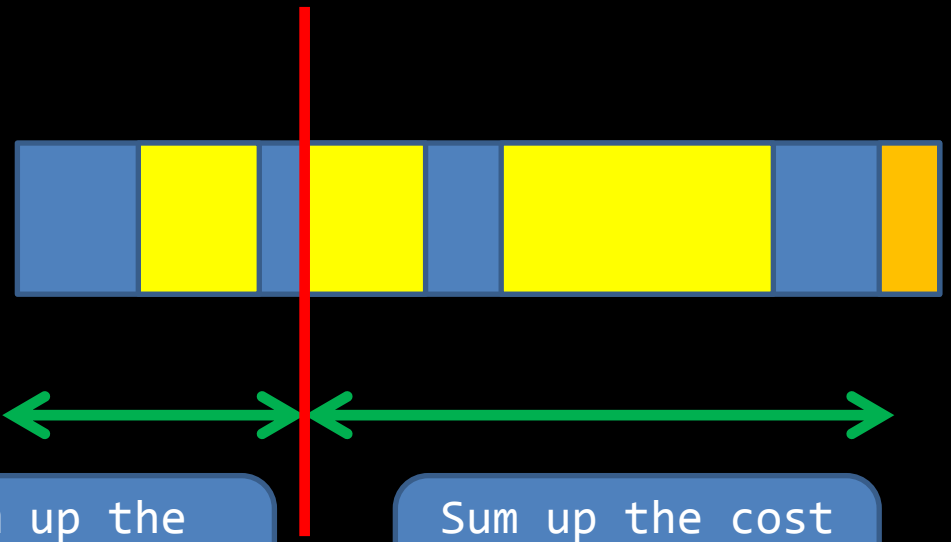
Algorithm 1

- Try all coding session and fix it. O(N)
  - Left side: try all possible position for the line. O(N)
    - Sum up the cost.
  - Choose the best line O(N)
  - Right side: do the same thing.
- Choose the best one at all.

- Complexity: $O(N^3)$
- Works for N up to 100, 25 marks

Speed up

Adding is too slow!



Sum up the cost of coding session

+

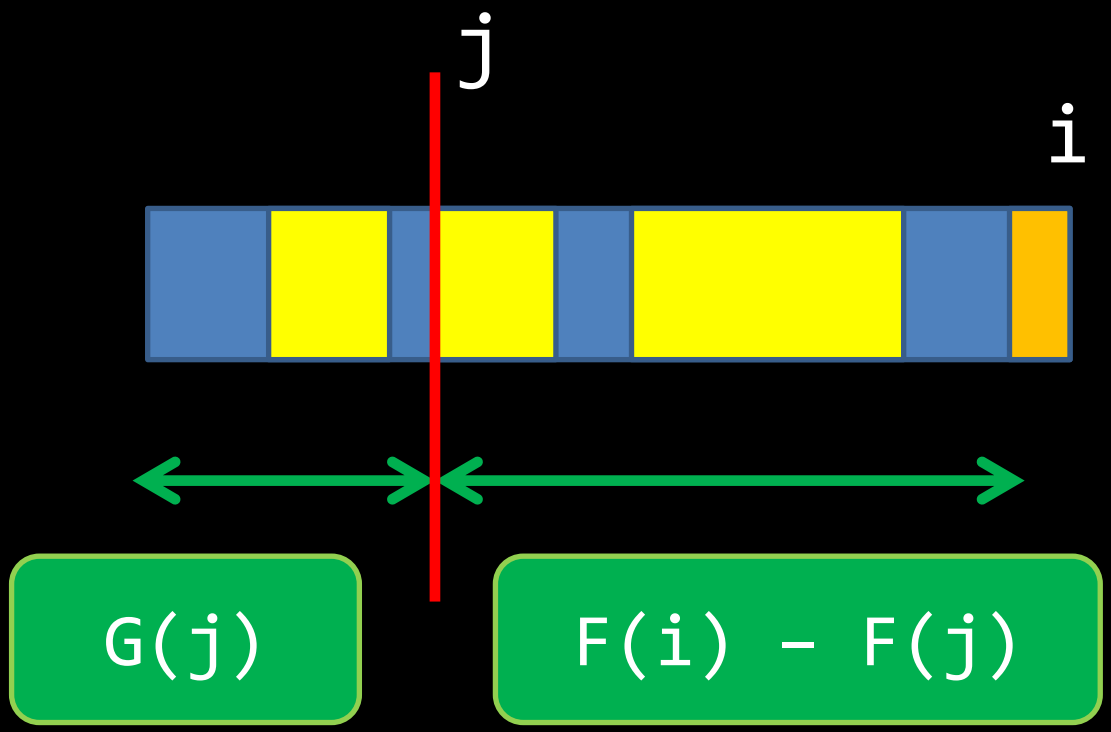
Sum up the cost of non-coding session

Do it in O(1)?

Speed up

- F(i) = Sum of cost of coding session in 1, 2, 3 .., i
- G(i) = Sum of cost of coding session in 1, 2, 3 .., i
- Compute F() and G() at the beginning

Speed up



Algorithm 2

- Compute F() and G() O(N)
- Try all coding session and fix it. O(N)
  - Left side: try all possible position for the line. O(N)
    - Compute the cost O(1)
  - Choose the best line
  - Right side: do the same thing.
- Choose the best one at all.

- Complexity: $O(N^2)$
- Works for N up to 3000, 50 marks

Speed up

- What we are doing
  - For each i, find j (j < i) such that F(i) – F(j) + G(j) is smallest

Define H(j) = F(j) – G(j)

  - For each i, find j (j < i) such that F(i) – H(j) is smallest

Speed up

- Finding greatest H(j) for j < i
- Pre-compute
  - opt(i) = greatest H(j) for j < i
  - opt(i) = max(opt(i-1), H(i-1))

Cost = F(i) + opt(i)

Algorithm 3

- Compute F(), G(), H() and opt() O(N)
- Try all coding session and fix it. O(N)
    - Compute the optimal cost for left side O(1)
    - Compute the optimal cost for right side O(1)
- Choose the best one at all.

- Complexity: O(N)
- Works for N up to 500000, 100 marks

Wait..

How about the cross midnight scenario?

Think in another way

Grouping Coding Session = Grouping non-Coding Session