HKOI Senior Q3 (Desktop Icons)
Editorial

Alex Tung
alex20030190@yahoo.com.hk

27 January 2018
1 Task Description

2 Statistics and Comments

3 Solution
   - Overview and simple cases
   - Part 1: finding optimal value and wallpaper
   - Part 2: finding a sequence of moves
Task Description

- Given $N$ icons on a $R \times C$ desktop. Each icon is either black (B) or white (W) and occupies one slot.

- Each icon also has an importance value $v_i$.

- Given $K$ wallpapers. Each wallpaper is black-and-white, with $R \times C$ tiles.

- A slot color = icon color $\rightarrow$ icon hidden;

- A slot color $\neq$ icon color $\rightarrow$ icon visible.

Your task: to maximize the sum of importance values of visible icons.

You may choose a wallpaper and move the icons around.

Perform the following at most 6400 times: choose an icon and move it to an empty slot.
Task Description

- Given $N$ icons on a $R \times C$ desktop. Each icon is either black (B) or white (W) and occupies one slot.
- Each icon also has an importance value $v_i$.
Given $N$ icons on a $R \times C$ desktop. Each icon is either black (B) or white (W) and occupies one slot.

Each icon also has an importance value $v_i$.

Given $K$ wallpapers. Each wallpaper is black-and-white, with $R \times C$ tiles.
Task Description

- Given $N$ icons on a $R \times C$ desktop. Each icon is either black (B) or white (W) and occupies one slot.
- Each icon also has an importance value $v_i$.
- Given $K$ wallpapers. Each wallpaper is black-and-white, with $R \times C$ tiles.
- $\text{slot\_color} = \text{icon\_color} \rightarrow \text{icon hidden}$;
- $\text{slot\_color} \neq \text{icon\_color} \rightarrow \text{icon visible}$.
Task Description

- Given $N$ icons on a $R \times C$ desktop. Each icon is either black (B) or white (W) and occupies one slot.
- Each icon also has an importance value $v_i$.
- Given $K$ wallpapers. Each wallpaper is black-and-white, with $R \times C$ tiles.
- $\text{slot}\_\text{color} = \text{icon}\_\text{color} \rightarrow \text{icon hidden};$
- $\text{slot}\_\text{color} \neq \text{icon}\_\text{color} \rightarrow \text{icon visible}.$
- Your task: to maximize the sum of importance values of visible icons.
Task Description

- Given $N$ icons on a $R \times C$ desktop. Each icon is either black (B) or white (W) and occupies one slot.
- Each icon also has an importance value $v_i$.
- Given $K$ wallpapers. Each wallpaper is black-and-white, with $R \times C$ tiles.
- $\text{slot\_color} = \text{icon\_color} \rightarrow \text{icon hidden}$;
- $\text{slot\_color} \neq \text{icon\_color} \rightarrow \text{icon visible}$.
- Your task: to maximize the sum of importance values of visible icons.
- You may choose a wallpaper and move the icons around.
Task Description

- Given $N$ icons on a $R \times C$ desktop. Each icon is either black (B) or white (W) and occupies one slot.
- Each icon also has an importance value $v_i$.
- Given $K$ wallpapers. Each wallpaper is black-and-white, with $R \times C$ tiles.

  - $\text{slot}_\text{color} = \text{icon}_\text{color} \rightarrow \text{icon hidden}$;
  - $\text{slot}_\text{color} \neq \text{icon}_\text{color} \rightarrow \text{icon visible}$.

- Your task: to maximize the sum of importance values of visible icons.
- You may choose a wallpaper and move the icons around.
- Perform the following at most 6400 times: choose an icon and move it to an empty slot.
Sample IO

Sample Input 1
1 2 2
1 1 W 10
1
WW
WW

Sample Output 1
0 1 0

Sample Input 2
1 2 2
1 1 W 10
2
WW
WW
WW
BW

Sample Output 2
10 2 1
1 1 2 1
Sample IO

Sample Input 3
3 2 2
1 1 B 10
1 2 W 10
2 2 B 10
1
BW
WB

Sample Output 3
30 1 3
1 1 2 1
1 2 1 1
2 2 1 2

Sample Output 3b
30 1 0
Constraints

For all cases:
1 ≤ N ≤ R × C
1 ≤ R, C ≤ 80
0 ≤ v_i ≤ 10^5
1 ≤ K ≤ 100

- Correct sum and sequence of (at most 6400) moves: 100%
- Correct sum: 40%

<table>
<thead>
<tr>
<th>Points</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>R, C ≤ 40</td>
</tr>
<tr>
<td></td>
<td>K = 1</td>
</tr>
<tr>
<td></td>
<td>All wallpaper tiles are of the same color</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>R, C ≤ 40</td>
</tr>
<tr>
<td></td>
<td>K = 1</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>R, C ≤ 40</td>
</tr>
<tr>
<td></td>
<td>N = R × C</td>
</tr>
<tr>
<td>4</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td>R, C ≤ 40</td>
</tr>
<tr>
<td></td>
<td>v_i = 1</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>R, C ≤ 40</td>
</tr>
<tr>
<td>6</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>No additional constraints</td>
</tr>
</tbody>
</table>
Statistics

Attempts: 48
Mean: 21.916
Stddev: 22.272
Top scores: 100 (dbsgame, 1:27), 69.4 (dbscat), 56.8 (6 contestants)
Score distribution:
One of the hardest tasks in HKOI Senior
One of the hardest tasks in HKOI Senior

Algorithmically simple, but very hard to code...
• One of the hardest tasks in HKOI Senior
• Algorithmically simple, but very hard to code...
• Risky to attempt full solution in the beginning
Subtask 1 (12 points): $R, C \leq 40, K = 1$, tiles of same color
Subtask 1 (12 points): \( R, C \leq 40, K = 1 \), tiles of same color

- No need to move icons.
Subtask 1 (12 points): \( R, C \leq 40, K = 1 \), tiles of same color

- No need to move icons.
- Just sum the values of all icons, whose color is different from that of the tiles.
Subtask 3 (16 points): \( R, C \leq 40, N = R \times C \)
Subtask 3 (16 points): \( R, C \leq 40, N = R \times C \)

- Cannot move icons.
Subtask 3 (16 points): $R, C \leq 40, N = R \times C$

- Cannot move icons.
- For each wallpaper, it is straightforward to calculate the sum of values of visible icons.
Subtasks 2 and 4

Subtask 2 (15 points): $R, C \leq 40, K = 1$
Subtask 4 (21 points): $R, C \leq 40, v_i = 1$
Subtasks 2 and 4

Subtask 2 (15 points): $R, C \leq 40, K = 1$
Subtask 4 (21 points): $R, C \leq 40, v_i = 1$

- Solving any of these two subtasks is not much easier than solving subtask 5 ($R, C \leq 40$)... It's just easier to code.
Subtasks 5 and 6

Subtask 5 (12 points): $R, C \leq 40$
Subtask 6 (24 points): No additional constraints ($R, C \leq 80$)
Subtasks 5 and 6

Subtask 5 (12 points): \( R, C \leq 40 \)
Subtask 6 (24 points): No additional constraints \( (R, C \leq 80) \)

- There are two parts to the solution.
Subtasks 5 and 6

Subtask 5 (12 points): $R, C \leq 40$
Subtask 6 (24 points): No additional constraints ($R, C \leq 80$)

- There are two parts to the solution.
- Part 1: to calculate the optimal value and find the wallpaper to be used.
Subtasks 5 and 6

Subtask 5 (12 points): \( R, C \leq 40 \)
Subtask 6 (24 points): No additional constraints \( (R, C \leq 80) \)

- There are two parts to the solution.
- Part 1: to calculate the optimal value and find the wallpaper to be used.
- Part 2: to find a sequence of moves to achieve the optimal value.
Subtasks 5 and 6

Subtask 5 (12 points): \( R, C \leq 40 \)
Subtask 6 (24 points): No additional constraints \( (R, C \leq 80) \)

- There are two parts to the solution.
- Part 1: to calculate the optimal value and find the wallpaper to be used.
- Part 2: to find a sequence of moves to achieve the optimal value.
- The intended solutions for subtasks 5 and 6 differ only in part 2.
Part 1: finding optimal value and wallpaper

Assumption

Assume that $N < R \times C$. 

For a wallpaper, we only need to care about the number of black (and white) tiles.

Suppose that a given wallpaper has $B$ black tiles and $W$ white tiles, and we have $b$ black icons and $w$ white icons. Clearly we should choose the $\min(b, W)$ black icons and the $\min(w, B)$ white icons of the highest values. Call these chosen icons good and other icons bad.
Assumption

Assume that $N < R \times C$.

- For a wallpaper, we only need to care about the number of black (and white) tiles.
Part 1: finding optimal value and wallpaper

Assumption

Assume that $N < R \times C$.

- For a wallpaper, we only need to care about the number of black (and white) tiles.
- Suppose that a given wallpaper has $B$ black tiles and $W$ white tiles, and we have $b$ black icons and $w$ white icons.
Part 1: finding optimal value and wallpaper

Assumption

Assume that $N < R \times C$.

- For a wallpaper, we only need to care about the number of black (and white) tiles.
- Suppose that a given wallpaper has $B$ black tiles and $W$ white tiles, and we have $b$ black icons and $w$ white icons.
- Clearly we should choose the min($b$, $W$) black icons and the min($w$, $B$) white icons of the highest values.
Part 1: finding optimal value and wallpaper

Assumption

Assume that \( N < R \times C \).

- For a wallpaper, we only need to care about the number of black (and white) tiles.
- Suppose that a given wallpaper has \( B \) black tiles and \( W \) white tiles, and we have \( b \) black icons and \( w \) white icons.
- Clearly we should choose the \( \min(b, W) \) black icons and the \( \min(w, B) \) white icons of the highest values.
- Call these chosen icons \textit{good} and other icons \textit{bad}. 
Part 2: finding a sequence of moves

Now we have chosen a wallpaper and know which icons are good. We want to make all good icons visible.
Part 2: finding a sequence of moves

- Now we have chosen a wallpaper and know which icons are good.
Part 2: finding a sequence of moves

- Now we have chosen a wallpaper and know which icons are good.
- We want to make all good icons visible.
Algorithm 1: Icon Swapping

Each time, we “swap” two icons using three moves.
Algorithm 1: Icon Swapping

Each time, we “swap” two icons using three moves.

Pseudo-code

1. while there is a hidden good icon
   
   Set $GOOD :=$ the hidden good icon
   
   Find $EMPTY :=$ an empty slot
   
   Find $TARGET :=$ a tile of opposite color to $GOOD$ and not occupied by a visible good icon
   
   Perform $GOOD \rightarrow EMPTY$
   
   Perform $TARGET \rightarrow GOOD$
   
   Perform $EMPTY \rightarrow TARGET$

end
Algorithm 1: Icon Swapping

Each time, we “swap” two icons using three moves.

Pseudo-code

while there is a hidden good icon
    Set $GOOD :=$ the hidden good icon
    Find $EMPTY :=$ an empty slot
    Find $TARGET :=$ a tile of opposite color to $GOOD$ and not occupied by a visible good icon
    Perform $GOOD \rightarrow EMPTY$
    Perform $TARGET \rightarrow GOOD$
    Perform $EMPTY \rightarrow TARGET$
end

The number of moves is at most $3N$, which solves subtasks 1 - 5.
Algorithm 1: Icon Swapping

Each time, we “swap” two icons using three moves.

Pseudo-code

while there is a hidden good icon
    Set $GOOD :=$ the hidden good icon
    Find $EMPTY :=$ an empty slot
    Find $TARGET :=$ a tile of opposite color to $GOOD$ and not occupied by a visible good icon
    Perform $GOOD \rightarrow EMPTY$
    Perform $TARGET \rightarrow GOOD$
    Perform $EMPTY \rightarrow TARGET$
end

The number of moves is at most $3N$, which solves subtasks 1 - 5.

Time complexity: $O(NRC)$. 
Algorithm 2: Greedily Fix Icons

- For ease of analysis, for each icon (good or bad) we fix a target colour.
Algorithm 2: Greedily Fix Icons

- For ease of analysis, for each icon (good or bad) we fix a target colour.
- At the end, all icons should be placed in a slot having the target colour.
Algorithm 2: Greedily Fix Icons

- For ease of analysis, for each icon (good or bad) we fix a target colour.
- At the end, all icons should be placed in a slot having the target colour.

Pseudo-code

1. while there is a misplaced icon that can be fixed
   Let \text{ICON} := \text{the misplaced icon}
   Let \text{TARGET} := \text{an empty slot having the target colour}
   Perform \text{ICON} \rightarrow \text{TARGET}

Each move, we fix exactly one misplaced icon, so the number of moves is at most \( N \). Accepted!
Algorithm 2: Greedily Fix Icons

- For ease of analysis, for each icon (good or bad) we fix a target colour.
- At the end, all icons should be placed in a slot having the target colour.

Pseudo-code

1. while there is a misplaced icon that can be fixed
   Let $ICON :=$ the misplaced icon
   Let $TARGET :=$ an empty slot having the target colour
   Perform $ICON \rightarrow TARGET$

Each move, we fix exactly one misplaced icon, so the number of moves is at most $N$. Accepted!

Time complexity: $O(NRC)$. 
The End

Questions?