M — Marvelous Marathon

Problem

Find a path of length x subject to certain constraints in a $2 \times m$ grid so that the sum of the values in the cells of the path is maximized.

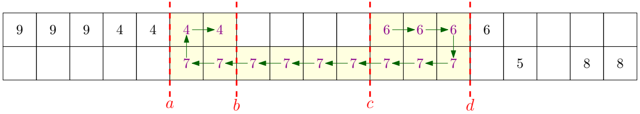
M — Marvelous Marathon

Problem

Find a path of length x subject to certain constraints in a $2 \times m$ grid so that the sum of the values in the cells of the path is maximized.

Formalized version of problem

Find integers $0 \leq a \leq b \leq c \leq d \leq m$ such that $2(b-a)+(c-b)+2(d-c)=x$.



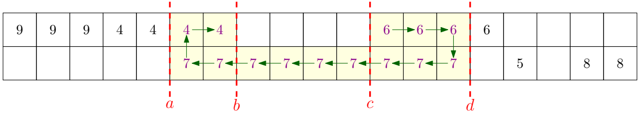
Problem

Find a path of length x subject to certain constraints in a $2 \times m$ grid so that the sum of the values in the cells of the path is maximized.

Formalized version of problem

Find integers $0 \leq a \leq b \leq c \leq d \leq m$ such that $2(b-a)+(c-b)+2(d-c)=x$.

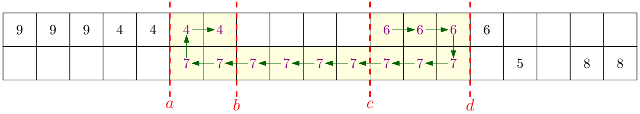
$$\text{Maximize} \left\{ \begin{array}{c} \text{sum of cells of one row in range } [a,d) \\ \text{plus} \\ \text{sum of cells of other row in ranges } [a,b) \text{ and } [c,d) \end{array} \right\}$$



M — Marvelous Marathon

Solution outline

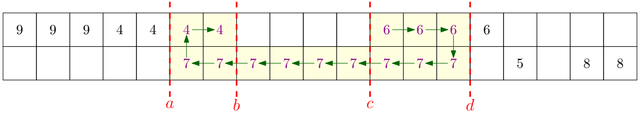
1. m is very large, cannot loop over all cells



M — Marvelous Marathon

Solution outline

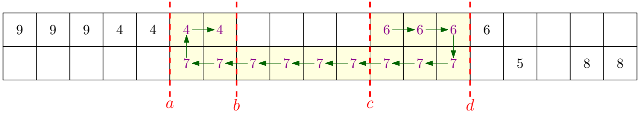
1. m is very large, cannot loop over all cells
2. Large parts of grid look the same because only n segments



M — Marvelous Marathon

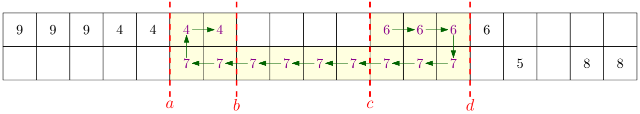
Solution outline

1. m is very large, cannot loop over all cells
2. Large parts of grid look the same because only n segments
3. Separately handle three main cases: 0, 1 or 2 U-turns



Solution outline

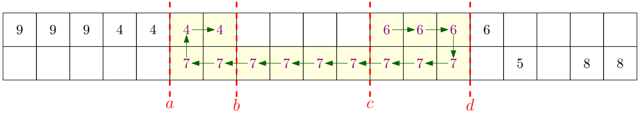
1. m is very large, cannot loop over all cells
2. Large parts of grid look the same because only n segments
3. Separately handle three main cases: 0, 1 or 2 U-turns
4. We focus here only on the hardest case with 2 U-turns.



M — Marvelous Marathon

Insight 1

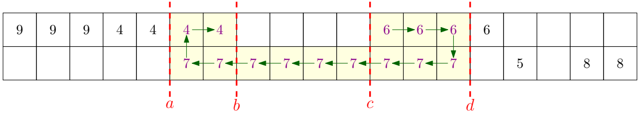
1. We can assume solution has the gap in the lower half
   - Run solution again on flipped input to cover opposite case



M — Marvelous Marathon

Insight 1

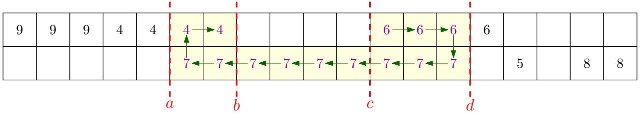
1. We can assume solution has the gap in the lower half
   - Run solution again on flipped input to cover opposite case
2. There is an optimal solution where a or d is at a segment endpoint (or 0 or m). Otherwise we could decrease (or increase) both a and d with 1 until either



M — Marvelous Marathon

Insight 1

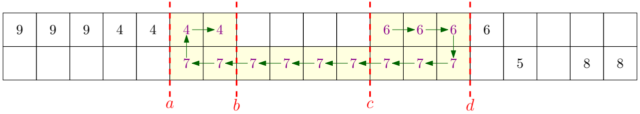
1. We can assume solution has the gap in the lower half
   - Run solution again on flipped input to cover opposite case
2. There is an optimal solution where a or d is at a segment endpoint (or 0 or m). Otherwise we could decrease (or increase) both a and d with 1 until either
   - a or d reaches a segment endpoint, *or*



M — Marvelous Marathon

Insight 1

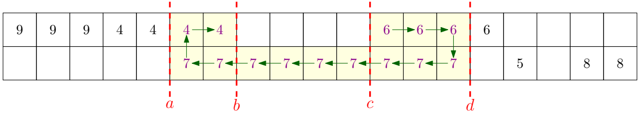
1. We can assume solution has the gap in the lower half
   - Run solution again on flipped input to cover opposite case
2. There is an optimal solution where a or d is at a segment endpoint (or 0 or m). Otherwise we could decrease (or increase) both a and d with 1 until either
   - a or d reaches a segment endpoint, *or*
   - $d = c$ or $a = b$, in which case we end up with the 1 U-turn case (handled separately, left as an exercise!)



M — Marvelous Marathon

Insight 1

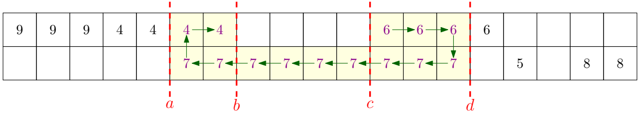
1. We can assume solution has the gap in the lower half
   - Run solution again on flipped input to cover opposite case
2. There is an optimal solution where a or d is at a segment endpoint (or 0 or m). Otherwise we could decrease (or increase) both a and d with 1 until either
   - a or d reaches a segment endpoint, *or*
   - $d = c$ or $a = b$, in which case we end up with the 1 U-turn case (handled separately, left as an exercise!)
3. We can assume a is the endpoint
   - Run solution again on reversed input to cover opposite case.



M — Marvelous Marathon

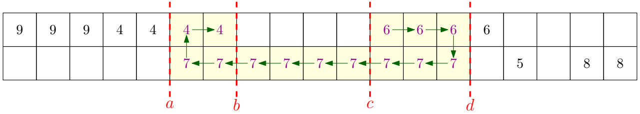
Insight 2

1. There is an optimal solution where b or c is at a segment endpoint, for the same reasoning as before (except that this time we would show it by shifting b or c).



Insight 2

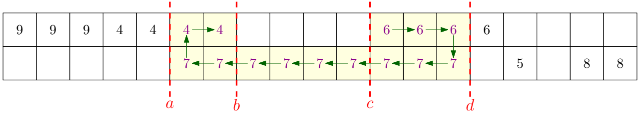
1. There is an optimal solution where b or c is at a segment endpoint, for the same reasoning as before (except that this time we would show it by shifting b or c).
2. We end up with two cases to consider:
   - a and b are segment endpoints
   - a and c are segment endpoints



M — Marvelous Marathon

Insight 2

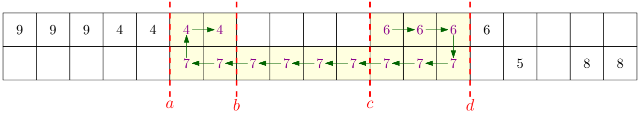
1. There is an optimal solution where b or c is at a segment endpoint, for the same reasoning as before (except that this time we would show it by shifting b or c).
2. We end up with two cases to consider:
   - a and b are segment endpoints
   - a and c are segment endpoints
3. Will focus on the first of these; the other must also be solved, but is done in a very similar fashion.



M — Marvelous Marathon

Sliding

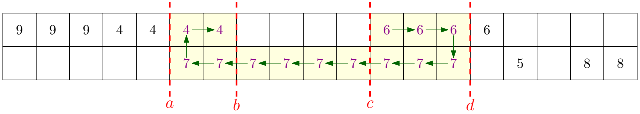
1. Fix some a and b. ($O(n^2)$ possible choices.)



M — Marvelous Marathon

Sliding

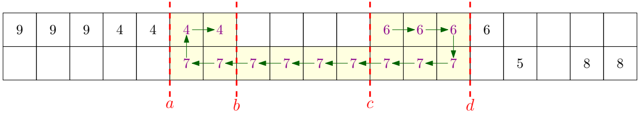
1. Fix some a and b. ($O(n^2)$ possible choices.)
2. Set $c = b$ (or $c = b + 1$ if x is odd) and $d = a + \lceil x/2 \rceil$



M — Marvelous Marathon

Sliding

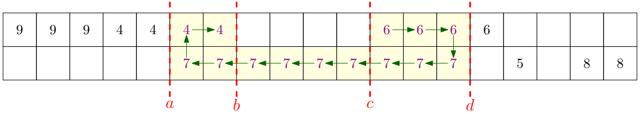
1. Fix some a and b. ($O(n^2)$ possible choices.)
2. Set $c = b$ (or $c = b + 1$ if x is odd) and $d = a + \lceil x/2 \rceil$
3. Idea: slide c and d right (c twice as fast as d) and maintain current score.



M — Marvelous Marathon

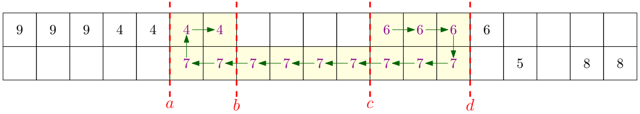
Sliding

1. Fix some a and b. ($O(n^2)$ possible choices.)
2. Set $c = b$ (or $c = b + 1$ if x is odd) and $d = a + \lceil x/2 \rceil$
3. Idea: slide c and d right (c twice as fast as d) and maintain current score.
   - Since grid values rarely change value, we can slide many steps at a time.



Sliding

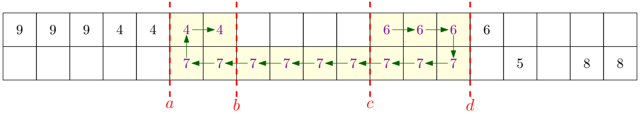
1. Fix some a and b. ($O(n^2)$ possible choices.)
2. Set $c = b$ (or $c = b + 1$ if x is odd) and $d = a + \lceil x/2 \rceil$
3. Idea: slide c and d right (c twice as fast as d) and maintain current score.
   - Since grid values rarely change value, we can slide many steps at a time.
   - Calculate when either c or d hits next segment endpoint and jump directly there.



M — Marvelous Marathon

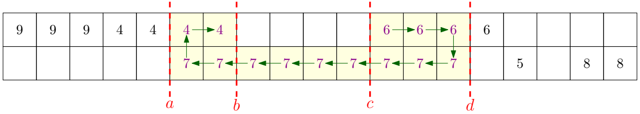
Sliding

1. Fix some a and b. ($O(n^2)$ possible choices.)
2. Set $c = b$ (or $c = b + 1$ if x is odd) and $d = a + \lceil x/2 \rceil$
3. Idea: slide c and d right (c twice as fast as d) and maintain current score.
    - Since grid values rarely change value, we can slide many steps at a time.
    - Calculate when either c or d hits next segment endpoint and jump directly there.
    - Repeat until c reached the end of the road.



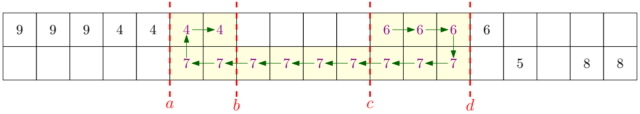
Sliding

1. Fix some a and b. ($O(n^2)$ possible choices.)
2. Set $c = b$ (or $c = b + 1$ if x is odd) and $d = a + \lceil x/2 \rceil$
3. Idea: slide c and d right (c twice as fast as d) and maintain current score.
   - Since grid values rarely change value, we can slide many steps at a time.
   - Calculate when either c or d hits next segment endpoint and jump directly there.
   - Repeat until c reached the end of the road.
4. "Next segment endpoint" can be found in $O(1)$, for a total complexity of $O(n^3)$.



Sliding

1. Fix some a and b. ($O(n^2)$ possible choices.)
2. Set $c = b$ (or $c = b + 1$ if x is odd) and $d = a + \lceil x/2 \rceil$
3. Idea: slide c and d right (c twice as fast as d) and maintain current score.
   - Since grid values rarely change value, we can slide many steps at a time.
   - Calculate when either c or d hits next segment endpoint and jump directly there.
   - Repeat until c reached the end of the road.
4. "Next segment endpoint" can be found in $O(1)$, for a total complexity of $O(n^3)$.
   - An optimized $O(n^4)$ implementation might also pass.



M — Marvelous Marathon

Sliding

1. Fix some a and b. ($O(n^2)$ possible choices.)
2. Set $c = b$ (or $c = b + 1$ if x is odd) and $d = a + \lceil x/2 \rceil$
3. Idea: slide c and d right (c twice as fast as d) and maintain current score.
   - Since grid values rarely change value, we can slide many steps at a time.
   - Calculate when either c or d hits next segment endpoint and jump directly there.
   - Repeat until c reached the end of the road.
4. "Next segment endpoint" can be found in $O(1)$, for a total complexity of $O(n^3)$.
   - An optimized $O(n^4)$ implementation might also pass.

Statistics at 4-hour mark: 0 submissions, 0 accepted