COMP4128 Week 02 Tutorial

Yifan He

z5173587@unsw.edu.au

https://github.com/hharryyf/COMP4128-24T3-tutoring

Outline

- Substring Removal Game
- Quality of Living
- Hints of problem set 1 by email

Substring Removal Game

Alice and Bob play a game. They have a binary string S ($|S| \leq 100$). Alice moves first, then Bob, then Alice again... During their move, the player can choose any positive number of consecutive equal characters in S and delete them. The game ends when the string becomes empty, and the score of each player is the number of 1-characters deleted by them. Each player wants to maximize their score. Calculate the resulting score of Alice. ($1 \leq T \leq 500$)

Example

S = 0110111110111, answer is 6.

Substring Removal Game

Observation 1

- Do we need to remove any 0?
- S = 011011111110111

Substring Removal Game

Observation 1

- Do we need to remove any 0?
- S = 011011111110111
- No! We'll remain more 1 for our opponent, and we'll group the 1 together.

Substring Removal Game

Observation 1

- Do we need to remove any 0?
- S = 011011111110111
- No! We'll remain more 1 for our opponent, and we'll group the 1 together.

Observation 2

- For a block of 1, can we remove part of it?
- S = 011011111110111

Substring Removal Game

Observation 1

- Do we need to remove any 0?
- S = 011011111110111
- No! We'll remain more 1 for our opponent, and we'll group the 1 together.

Observation 2

- For a block of 1, can we remove part of it?
- S = 011011111110111
- No! We'll remain more 1 for our opponent, and our opponent can remove a consecutive block.

Substring Removal Game

Algorithm

- Any ideas?

Substring Removal Game

Algorithm

- Any ideas?
- Alice grabs the largest consecutive block of 1, Bob grabs the second largest consecutive block of 1, Alice grabs the third largest consecutive block of 1...
- S = 011011111011, answer is 6.

Demo

Quality of Living

Given a matrix of size $R \times C$ that contains all numbers between 1 and $R \cdot C$. Find a submatrix of size $H \times W$ that has the smallest median.

$1 \leq R, C \leq 3000$, $1 \leq H \leq R$, $1 \leq H \leq C$; H, W are **odd**.

Example

5	11	12	16	25
17	18	2	7	10
4	23	20	3	1
24	21	19	14	9
6	22	8	13	15

Quality of Living

Naive approach 1

- Calculate the median of all submatrices of size $H \times W$
- For every submatrix of size $H \times W$, find the $\lceil \frac{H \cdot W}{2} \rceil$-th largest element m, find the minimum of such m among all submatrices.
- Total number of submatrices: $O(R \cdot C)$
- Time required to find the $\lceil \frac{H \cdot W}{2} \rceil$-th largest element: Average $O(H \cdot W)$ (using quicksort)
- Time complexity: $O(R^2 \cdot C^2)$

Quality of Living

Analysis 1

- Median calculation is slow

Quality of Living

Analysis 1

- Median calculation is slow
- Every submatrix, the median calculation complexity is linear

Quality of Living

Analysis 1

- Median calculation is slow
- Every submatrix, the median calculation complexity is linear
- Calculating the median for every submatrix separately doesn't work!

Quality of Living

Analysis 2

- Finding the median of every submatrix directly won't work

Quality of Living

Analysis 2

- Finding the median of every submatrix directly won't work
- We don't really need the median of all submatrices, just the smallest one

Quality of Living

Analysis 2

- Finding the median of every submatrix directly won't work
- We don't really need the median of all submatrices, just the smallest one
- Suppose the optimal answer is K, there must exist a submatrix such that its median is **no greater than** K.

Quality of Living

Analysis 2

- Finding the median of every submatrix directly won't work
- We don't really need the median of all submatrices, just the smallest one
- Suppose the optimal answer is K, there must exist a submatrix such that its median is **no greater than** K.
- Can we check if there exists a submatrix of size $H \cdot W$ such that the median is **no greater than** K?

Quality of Living

Analysis 2

- Finding the median of every submatrix directly won't work
- We don't really need the median of all submatrices, just the smallest one
- Suppose the optimal answer is K, there must exist a submatrix such that its median is **no greater than** K.
- Can we check if there exists a submatrix of size $H \cdot W$ such that the median is **no greater than** K?
- Desired time complexity $O(R \cdot C)$

Quality of Living

Critical Observation

- What is median?

Quality of Living

Critical Observation

- What is median?
- The middle element

Quality of Living

Critical Observation

- What is median?
- The middle element
- How to know the median is no greater than K for $H \cdot W$ many elements?

Quality of Living

Critical Observation

- What is median?
- The middle element
- How to know the median is no greater than K for $H \cdot W$ many elements?
- There are at least $\lceil \frac{H \cdot W}{2} \rceil$ of the elements no greater than K!

Quality of Living

Critical Observation

- What is median?
- The middle element
- How to know the median is no greater than K for $H \cdot W$ many elements?
- There are at least $\lceil \frac{H \cdot W}{2} \rceil$ of the elements no greater than K!
- If we replace every element no greater than K with +1, and every element greater than K with -1, then the median is no greater than K iff the submatrix has a positive sum!

Quality of Living

Example

Check if the colored submatrix has a median no greater than 9.

5	11	12	16	25
17	18	2	7	10
4	23	20	3	1
24	21	19	14	9
6	22	8	13	15

5	11	12	16	25
17	18	+1	+1	-1
4	23	-1	+1	+1
24	21	-1	-1	+1
6	22	8	13	15

Sum is 1, so yes!

Quality of Living

Algorithm

Objective: Check if there exists a submatrix of size $H \cdot W$ with median no more than K.

- Relace all numbers greater than K with -1, all numbers no more than K with +1

Quality of Living

Algorithm

Objective: Check if there exists a submatrix of size $H \cdot W$ with median no more than K.

- Relace all numbers greater than K with -1, all numbers no more than K with +1
- Find if there exists a submatrix of size $H \cdot W$ cumulative sum greater than 0.

Quality of Living

Algorithm

Objective: Check if there exists a submatrix of size $H \cdot W$ with median no more than K.

- Relace all numbers greater than K with -1, all numbers no more than K with +1
- Find if there exists a submatrix of size $H \cdot W$ cumulative sum greater than 0.
- Step 1: $O(R \cdot C)$

Quality of Living

Algorithm

Objective: Check if there exists a submatrix of size $H \cdot W$ with median no more than K.

- Relace all numbers greater than K with -1, all numbers no more than K with +1
- Find if there exists a submatrix of size $H \cdot W$ cumulative sum greater than 0.
- Step 1: $O(R \cdot C)$
- Step 2: $O(R \cdot C)$ with 2-d prefix-sum

Quality of Living

Preliminary

- You are given a 2-d matrix of size $O(R \cdot C)$
- You are also given $O(R \cdot C)$ preprocessing time
- How to answer queries like what is the sum of elements of some arbitrary submatrix in O(1)?

5	11	12	16	25
17	18	2	7	10
4	23	20	3	1
24	21	19	14	9
6	22	8	13	15

Quality of Living

Recall 1-d prefix sum

- Given an array A of size N
- Given O(N) preprocessing time
- Answer queries like $\sum_{i=l}^{r} A[i]$ in $O(1)$

Solution

- Create an 1-d array P of size N
- Let $P[i] = \sum_{i=1}^{i} A[i] = P[i-1] + A[i]$
- The above preprocessing step is $O(N)$
- Answer queries like $\sum_{i=l}^{r} A[i] = P[r] - P[l-1]$
- This is apparently $O(1)$ per query

Quality of Living

2-d prefix sum

- Create a 2-d array P of size $O(R \cdot C)$
- Let each element $P[i][j] = \sum_{1 \le k \le i, 1 \le l \le j} a[k][l]$.

2	7	10
20	3	1
19	14	9

2	9	19
22	32	43
41	65	85

Quality of Living

2-d prefix sum

- Create a 2-d array P of size $O(R \cdot C)$
- Let each element $P[i][j] = \sum_{1 \le k \le i, 1 \le l \le j} a[k][l]$.

2	7	10
20	3	1
19	14	9

2	9	19
22	32	43
41	65	85

Quick calculation of P[i][j]

- Dynamic Programming!

Quality of Living

2-d prefix sum

- Create a 2-d array P of size $O(R \cdot C)$
- Let each element $P[i][j] = \sum_{1 \le k \le i, 1 \le l \le j} a[k][l]$.

2	7	10
20	3	1
19	14	9

2	9	19
22	32	43
41	65	85

Quick calculation of P[i][j]

- Dynamic Programming!
- $P[i][j] = P[i-1][j] + P[i][j-1] - P[i-1][j-1] + a[i][j]$.

Quality of Living

Answer 2-d range sum query

- Let each element $P[i][j] = \sum_{1 \le r \le i, 1 \le c \le j} a[r][c]$.
- What is $\sum_{r_1 \le r \le r_2, c_1 \le c \le c_2} a[r][c]$?

Quality of Living

Answer 2-d range sum query

- Let each element $P[i][j] = \sum_{1 \le r \le i, 1 \le c \le j} a[r][c]$.
- What is $\sum_{r_1 \le r \le r_2, c_1 \le c \le c_2} a[r][c]$?
- $P[r_2][c_2] - P[r_2][c_2 - 1] - P[r_1 - 1][c_2] + P[r_1 - 1][c_1 - 1]$
- Obvious O(1) per query

Already lost? :)

Already lost? :)

Summary

Already lost? :)

Summary

- We want to check if there exists a submatrix of size $H \cdot W$ with median **no greater than** K.

Already lost? :)

Summary

- We want to check if there exists a submatrix of size $H \cdot W$ with median **no greater than** K.
- We change each element greater than K with -1, and each element no more than K with +1

Already lost? :)

Summary

- We want to check if there exists a submatrix of size $H \cdot W$ with median **no greater than** K.
- We change each element greater than K with -1, and each element no more than K with +1
- Equivalent to checking if there exists a submatrix of size $H \cdot W$ with cumulative sum greater than 0.

Already lost? :)

Summary

- We want to check if there exists a submatrix of size $H \cdot W$ with median **no greater than** K.
- We change each element greater than K with -1, and each element no more than K with +1
- Equivalent to checking if there exists a submatrix of size $H \cdot W$ with cumulative sum greater than 0.
- We can answer 2-d prefix range sum query with 2-d prefix sum and it is O(1) per query with $O(R \cdot C)$ preprocessing time.

Already lost? :)

Summary

- We want to check if there exists a submatrix of size $H \cdot W$ with median **no greater than** K.
- We change each element greater than K with -1, and each element no more than K with +1
- Equivalent to checking if there exists a submatrix of size $H \cdot W$ with cumulative sum greater than 0.
- We can answer 2-d prefix range sum query with 2-d prefix sum and it is O(1) per query with $O(R \cdot C)$ preprocessing time.
- Our objective is solved in $O(R \cdot C)$.

Quality of Living

Similar Feeling?

- We change a minimization problem to a validation problem

Quality of Living

Similar Feeling?

- We change a minimization problem to a validation problem
- We can check if the answer is no greater than K efficiently

Quality of Living

Similar Feeling?

- We change a minimization problem to a validation problem
- We can check if the answer is no greater than K efficiently
- If the answer is no greater than K, it is no greater than K+1

Quality of Living

Similar Feeling?

- We change a minimization problem to a validation problem
- We can check if the answer is no greater than K efficiently
- If the answer is no greater than K, it is no greater than K+1
- We need to find the smallest K such that the median of some submatrix is no greater than K

Quality of Living

Similar Feeling?

- We change a minimization problem to a validation problem
- We can check if the answer is no greater than K efficiently
- If the answer is no greater than K, it is no greater than K+1
- We need to find the smallest K such that the median of some submatrix is no greater than K
- Binary search!

Quality of Living

Similar Feeling?

- We change a minimization problem to a validation problem
- We can check if the answer is no greater than K efficiently
- If the answer is no greater than K, it is no greater than K+1
- We need to find the smallest K such that the median of some submatrix is no greater than K
- Binary search!
- Time complexity $O(R \cdot C \cdot log(R \cdot C))$

Quality of Living

The problem is very difficult, I had absolutely no idea when I first saw it. However, this problem is not a one-off example.

Related Problem

- Max Median
- `https://codeforces.com/contest/1486/problem/D`
- Match point
- `https://codeforces.com/contest/1156/problem/C`

Practice in your free time

Demo