Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №1-4

по «Алгоритмам и структурам данных»

Задачи Тимуса

Выполнил:

Студент группы P3233

Нгуен Нгок Дык

Преподаватели:

Косяков М.С.

Санкт-Петербург

2022

Code: https://github.com/ndwannafly/ITMO_ALGO

Задача № **1155. Troubleduons**

Task reformulation:

- Given a cube with 8 vertices labelled from A -> H.
- Each vertex u has a number called as a[u].
- To one edge of the cube, there are two types of operations:

  + Add one to both two vertices of the edge

  + Subtract one from each vertex of the edge
- Finding a solution to set all number of each vertex equal to 0.
- Number of operation must not be exceed 1000.

Constraint:

- 0 ≤ a[i] ≤ 100

Keyword: Math, greedy

Observation:

- **Observation 1**:

  + We can divide the vertices into two sets of vertices such that each operation will add one to each set or subtract one from each set. Easily to observe they are {A ; C ; H ; F} and { B ; D ; E; G }.

  + Obviously, it's "IMPOSSIBLE" if their sums are not equal.
- **Observation 2**:

  + We can move from A to C by using B as a transit point.

  For example:

  a -> b ->c

  a -> b +a -> c + a

  0 -> b -> c + a

Solution:

- **4 steps:**

– (1) Do subtraction operation for any two adjacent vertices which both have number > 0

– (2) Move all F , H , C to A (by applying observation 2)

– (3) Move all D, E, G to B (by applying observation 2)

– (4) Then continuously do subtraction operation for A and B

Complexity:

- We do less than 1000 operations.

Задача № **2025. Line Fighting**

Task reformulation:

- Divide n fighters into k teams such that number of the matches are maximum.
- There is match between any two fighters in two different team.

Constraint:

- K ≤ N ≤ 10000

Keyword: Greedy, Math

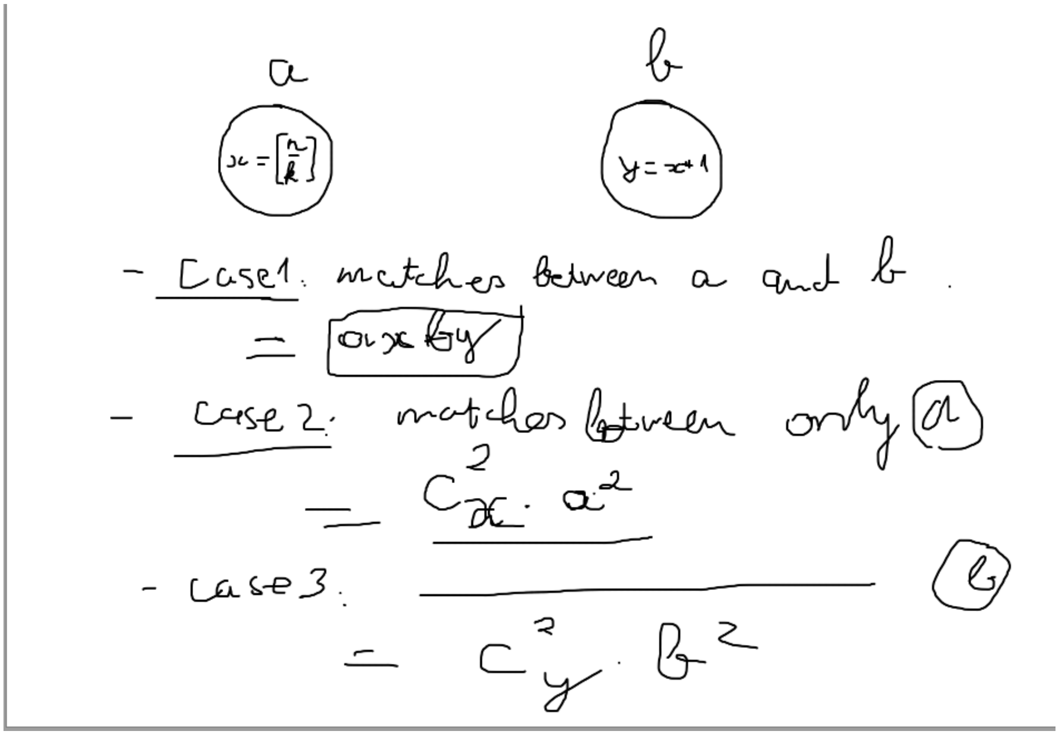
Solution:

- In other words, our mission is finding solution (x1, x2, x3, .. Xk) of equation x1 + x2 + x3 + ... + xk = n so that $\sum_{\substack{1\le i \le k \\ 1< j < k}} P(i,j)$ is maximum
- **Observation**: The sum is maximumm if the sizes of each set are equal.

  - **Proof**: I came up with a matimatical formula proof but this proof from discusson on the size is very nice.

Posted by John 14 Jan 2022 16:36

> So we must try to make teams have equal size. Why?
> Consider two teams with x and y members respectively, and consider x > y. Sending a member from first team to second team will only affect matches between these two teams. At first we have x*y different matches between the teams. If we send a player to the second team, there are now x-1 and y+1 members in each team. The matches are now
> (x-1)*(y+1) = x*y + (x-y) - 1
> As we said x > y, x-y > 0, so x-y-1 >= 0 and our change can't reduce the total number of matches, so it is an optimal change.
> This also shows that nothing changes when x = y+1, so if we can't make all teams equal size, just add each of the ones remaining into a different team, making a difference of at most 1 in the sizes of teams.
>
> Now it's your job to compute the number of matches, I won't show that, but it can be done with a one line formula. Good luck :)

- Initially, we will try to divide [n /k] fighters to each team.
- The remainders are distributed to each team.
- Therefore, We have a teams which have [n/k] fighters and b teams which have [n/k] + 1 fighters.
- 
a
b
x = [n/k]
y = x+1
- Case1: matches between a and b.
= axby
- case 2: matches between only a
= C_x^2 · a^2
- case3:
b
= C_y^2 · b^2
- Result = axby + C(2,x) * a^2 + C(2,y) * b^2

Complexity:

- Operations: O(1) for each testcase
- Spaces: O(1)

Задача № **1207. Median on the Plane**

Task reformulation:

- Given N (even) points (xi, yi) on the plane.
- No three points lie on the same straight line
- Find two points such that divide the plane into two parts so that each of them has equal points.

Constraint:

- $N \leq 10000$
- -10^6 ≤ xi, yi ≤ 10^6

Keyword: Geometry, 2D cross product, sort

Solution:

- First of all, because N is even and no three points lie on the same straight line, then there always exists a way to divide the plane into two equal-sized parts
- We fix a point as the origin, then sort the others by their 2D cross products to the fixed point.
- Read more about 2D cross product here **https://www.nagwa.com/en/explainers/175169159270/**
- Result is the fixed point and the median of the sorted array.

My comment:

- 2D cross product is a powerful tool to check if 3 points are collinear or clockwise moving (cw, ccw) when the two vectors have a common origin (this applies to the exercise of finding the **convex hull** of the point set)

Complexity:

- Operations: O(nlogn)
- Spacing: O(n)

Задача № **1604. Country of Fools**

Task reformulation:

- Given k elements in an array
- There are a[1] number 1, a[2] number 2, ... A[k] number k.
- A[1] + a[2] + a[3] + ... + a[k] = n
- Arrange these n numbers in such a way that maximize the times when a[i] != a[i-1]

Constraint:

- 1 ≤ k ≤ 10000
- 1 ≤ sum ai ≤ 10000

Keyword: Greedy, heap

Solution:

- Looping through the array, each time, the optimal solution is taking two maximum elements.
- Using a heap to manage the array.

Complexity:

- Operations: O(n log k)
- Spacing: O(n)

Задача № **1494. Monobilliards**

Task reformulation:

- A player is playing billard.
- He has to pocket successively the balls with numbers 1, 2, .. , N into the only pocket exactly in this order.
- Given the array is the order of getting out the ball check whether he is cheater.

Constraint:

- 1 ≤ N ≤ 100 000

Keyword: Data structures, set, lower_bound.

Solution:

- Looping through the array from the end, with each a[i], check whether or not on the right-hand side there is an element greater than a[i] but less than max(a[i+1] ... A[n])
- Persist a set, each time we use lower_bound function to finding element greater than a[i].
- Building an array R[i] = max (a[i+1] , ... , a[n])

Complexity:

- Operations: O(nlogn)
- Spacing: O(n)

Задача № **1628. White Streaks**

Task reformulation:

- Given a grid m x n.
- There are some blocked cells
- Paving the grid by using two types of tile: 1 x l or l x 1.
- Paving in such a way that there is no other way to paving a tile include the tile in the old way. (**no inclusion**)

Constraint:

- $1 \leq m, n \leq 100\ 000$
- $0 \leq k \leq 60000$

Keyword: Loop

Solution:

- Set all the outer-edge elements of the grid is black
- Let take two blacks in the **same row** or in the **same column:**
  - If there are **more than 1 space** between them, pave one tile fit in it
  - If there is **only one space** between them, we only pave if and only if this cell is isolated from all 4 sides.

Complexity:

- **Operations** : O(n * log n)
- **Spaces** : O(n)

Задача № **1450. Russian Pipelines**

Task reformulation:

- Given a graph N vertices M directed edges (arrows).
- Each edge has cost C[i]
- If there exists path from x to y, then no path from y to x.
- Given two vertices S and T
- Find the longest path from S to T

Constraint:

- $2 \leq N \leq 500$
- $0 \leq M \leq 124750$
- $1 \leq C[i] \leq 10000$.

My comment:

– We notice about this information « **If there exists path from x to y, then no path from y to x** »

– It means that there is **no cycle** in the graph, so finding longest path is possible by implementing Dijkstra's algorithm.

Keyword: Grpah, Dijkstra

Solution:

- Nothing special if we understand Dijkstra's algorithm. (https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm)

- I implemented dijkstra’s heap.

Complexity:

- **Operations** : O(V + MlogN)
- **Spaces** : O(M)

Задача № **1806. Mobile Telegraphs**

Task reformulation:

- Given N telegraphs, each of them is a string with digit from 0-9
- Given N-element array, c[1] ... c[n] are the costs
- Telegraph x can be sent to telegraph y if y can be obtained by **changing exactly one digit** or **swapping exactly two digits** in x.
- Cost for sending telegraph x to telegraph y is **c[the length of the longest common prefix of two strings]**
- Find shortest path from 1 to N

Constraint:

- 2 ≤ n ≤ 50000
- 1 ≤ c[i] ≤ 10000

My comment:

– The idea is very clear. Finding shortest path from 1 to N, Dijkstra is the best choice.

– Dijkstra’s algorithm complexity is about the number of edges in the graph.

– I got MLE because of making unnecessary edges.

– But after reducing them, I got AC with a **tight time**. So maybe there is other approach which can optimize the memory better then mine.

Keyword: Graph, dijkstra, data structures, implementation

Solution:

- Each telegraph :
  - Try to change each digit
  - Try to swap two any digits
  - If the achieved new string is another given string => make edge
- Dijkstra as usual from 1 to

Complexity:

- **Operations** : O(N + number of edge * log N)
- **Spaces** : depend on number of edges in the graph