· U C ·

UNIVERSIDADE DE COIMBRA
Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

Estratégias Algorítmicas

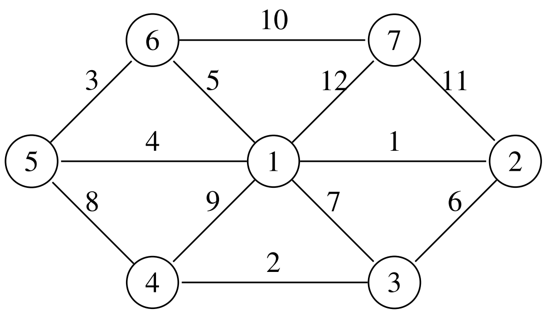
Exame Normal – 7 de junho de 2024

Nome: ______________________________ Nº de estudante: ______________

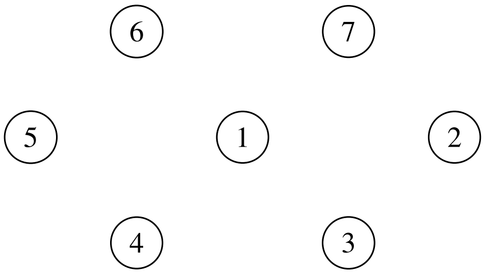
13 pontos no total, 2 horas, sem consulta.

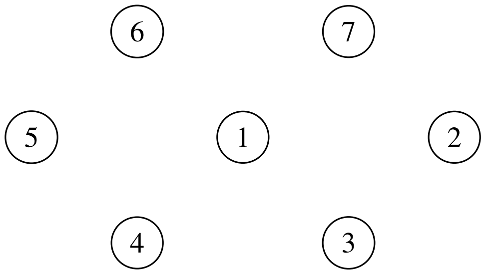
1. Escreva o pseudo-código de um algoritmo recursivo que, dado um conjunto de n elementos, imprima todos os subconjuntos que tenham mais do que $\lceil n/2 \rceil$ elementos. (1.5 pontos)

2. Considere o seguinte grafo.

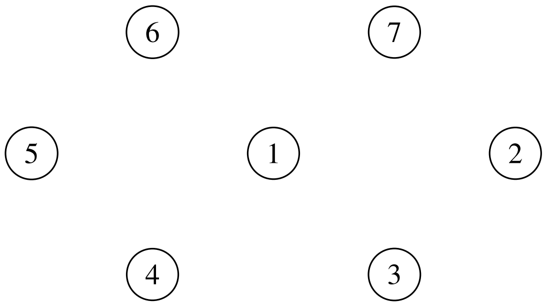


a) Desenhe a árvore geradora mínima (à esqueda) e o grafo da estrutura de dados *union-find*, sem o passo de compressão de caminho (à direita), recorrendo ao algoritmo de Kruskal. Quando necessário, ligue a raiz da árvore com menor altura à raiz da árvore com maior altura e, em caso de empate, escolha, como raiz, o vértice que apresentar a etiqueta com o menor valor. (2 pontos)

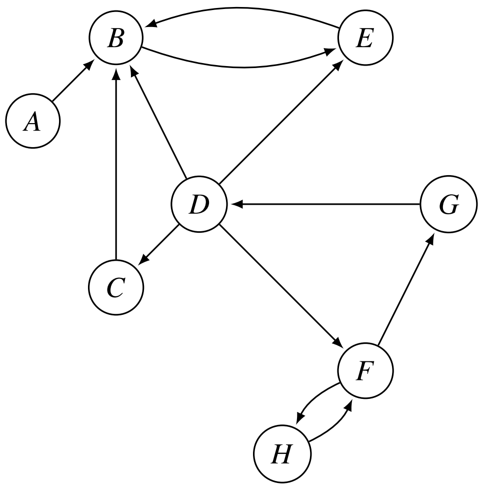




b) Qual seria a representação do grafo da estrutura de dados *union-find* se efetuasse o passo de compressão de caminho na última ligação efetuada pelo algoritmo de Kruskal? (0.5 pontos)



3. Encontre as componentes fortemente conexas do grafo seguinte. Para justificação da sua resposta, reporte a árvore de procura em profundidade do Algoritmo de Tarjan a partir do vértice A, escolhendo os vértices para a travessia de acordo com a ordem alfabética das etiquetas, e indique explicitamente os valores finais de `dfs` and `low` em cada vértice. Caso necessite de reiniciar o algoritmo noutro vértice, escolha aquele que tiver menor etiqueta (alfabéticamente). Reporte igualmente as componentes encontradas, ordenadas pelo tempo em que foram encontradas durante a travessia em profundidade (3 pontos)



4. Considere a seguinte recorrência para $i = 1, \ldots, n$, $j = 1, \ldots, n$ e $c_i \geq 0$:

$$C(i,j) = \begin{cases} 0 & \text{se } j < i \\ c_i & \text{se } i = j \\ \displaystyle\sum_{\ell=i}^{j} c_\ell + \min_{i \leq k \leq j} \{C(i,k-1) + C(k+1,j)\} & \text{caso contrário} \end{cases}$$

Tendo em conta que o valor que pretende obter é retornado por $C(1,n)$, apresente o pseudo-código de um algoritmo de programação dinâmica descendente (*top-down*) que explore a recorrência acima para obter esse valor e discuta a sua complexidade computacional. (3 pontos)

Nome: ____ Nº de estudante: _______________

5. Responda **unicamente** a duas das três questões seguintes sobre os três primeiros problemas de programação de Estruturas Algorítmicas. Uma versão mais compacta dos enunciados dos três problemas está disponível nas páginas seguintes.

   Indique de seguida quais as questões que são consideradas para avaliação (A, B, C): _________

A) **Problema A – Aztec Vaults** (1.5 pontos)
Considere o seguinte estado de uma instância do problema A:

```
1 1 1
2 3 3
3 2 2
```

Indique todos os próximos estados que serão visitados na próxima chamada recursiva, assumindo que pode efetuar os seguintes movimentos para cada *vault handle*:

`DIREITA, ESQUERDA, 2x DIREITA`

Deve indicar, para cada estado possível, o movimento que foi feito, as coordenadas i, j do *vault handle* que foi rodado (existem 4 com coordenadas $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$), e a matriz resultante. Caso a sua solução para o problema inclua condições de corte que excluem certos movimentos de serem feitos, deve indicar apenas as coordenadas (i,j) do *vault*, o movimento que excluiu e o corte que o levou a não considerar esse movimento.

Problem A – Aztec Vaults

Description

The year is 1950. You are a curious archaeology undergrad fascinated by the ancient culture of the South American people. Your teacher, Dr. Henry Jones, invites you on a trip to the depths of northern Mexico, in search of the lost Montezuma Treasure, an invitation that you gladly accept. For two weeks, you will meander in the lost cities of the Aztecs, fighting for survival and for your credit assignments that will allow you to graduate. Just when everything seems lost, you find your way into a hidden tunnel, filled with curious Aztec patterns.

At the end of the tunnel, you stumble into a wall, filled with vault handles. You have found the Vault to the Montezuma Treasure! As you start to celebrate, Dr. Jones warns you that this is not the moment to be happy. You are surrounded by skeletons of past treasure hunters, who failed to open the vault

The wall has several blocks, each with a pattern design. There is a vault handle in the middle of each 2x2 block grid. You play around with the vault handles, and you notice that when you rotate the handle, the surrounding 4 blocks rotate in the same direction.

As Dr. Jones inspects the surroundings, he finds a drawing of the correct configuration that unlocks the vault. That must mean you need to rearrange the blocks to fix the patterns in order to open the vault. Next to the drawing there is also a number and a skull. ”Hum, what do you make of this?” Dr. Jones asks you. ”Well sir, I would suspect that we have a limited number of moves we can make to open the vault, until the Aztec traps kill us like our friends here!”, you reply. ”I believe you are right. And you just wasted precious moves by fiddeling around with the handles!”.

Your mission is to rotate the vault handles in the minimum possible number of moves that restore it to its correct position! Note: The blocks are all considered the same if they share a pattern. What matters is that each pattern is in its corresponding line, i.e., pattern 1 must be in line 1 and so forth

Input: The input starts with one line containing one integer T that corresponds to the number of test cases. Then, T input blocks follow. Each input block starts with a line containing the number of rows R and columns C of the grid G, and the maximum number of moves allowed M, separated by whitespace. Then, R lines follow, each representing a row of the vault grid, containing C integers separated by whitespace.

Output: For each test case, you should print the minimum number of moves S required to restore the vault grid to its correct position in the format ”S”. If no solution is found, you should output ”the treasure is lost!”.

Nome: ______________________________ Nº de estudante: ____________

B) **Problema B – Another Aztec Ridle** (1.5 pontos)
Complete o seguinte pseudo-código para obter um algoritmo recursivo que imprima todas as matrizes M binárias com n linhas e n colunas e com exatamente ℓ 1s em cada coluna e em cada linha. Considere que a chamada recursiva preenche a posição correspondente à linha i e coluna j da matriz M, $i = 1, \ldots, n$ e $j = 1, \ldots, n$. As estruturas *row* e *col*, com n posições cada, guardam o número atual de 1s em cada linha e em cada coluna, respetivamente. Assuma que os valores de n e ℓ estão acessíveis por variáveis globais. Use a instrução $print(M)$ para imprimir a matriz quando estiver completa. Explique as estruturas de dados e variáveis que adicionar ao pseudo-código.

Function $gera(M, i, j, col, row)$
  if $i > n$ and $j > n$ **then**

    return
  else if $j > n$ **then**

    return
  else
    if $row[i] < \ell$ and $col[j] < \ell$ **then**
      $M[i][j] = 1$
      $row[i] = row[i] + 1$
      $col[j] = col[j] + 1$
      $gera(M, i, j+1, col, row)$

    $M[i][j] = 0$
    $gera(M, i, j+1, col, row)$

    return

Problem B - Another Aztec Riddle

Description

Once you found the minimum number of moves to open the vault, you and Dr. Henry Jones entered into a secret chamber. The Montezuma Treasure was finally in front of you, and you felt that particular shiver down the spine. What a satisfaction of seeing your dream come true! Both of you were completely flabbergasted by such amount of gold and gems.

Time to wake up! You started filling up your knapsack. You soon realized that the knapsack was not enough, so you decided to use your socks, your pockets, your hat, anything that could be used to collect the remaining pieces. But, suddenly, a deep voice was heard out of the chamber. ”Life is not that easy, there is still another riddle to be solved.”

And the door of the chamber closed. You felt again another shiver down the spine, but for a completely different reason. You soon realized that there was no possibility of getting out of the chamber. ”What can we do now?”, you asked to Dr. Jones, who shook his head and said, ”My son, I fear that this is not going to end well”. You started to feel desperate! You and Dr. Jones had eaten all the food during the trip, and there was very little drinking water in the bottle. And, of course, no one would hear you screaming for help inside the chamber. What a nightmare this has become!

As soon as you sat down over your knapsack, a flat object fell into your hands. What is this? ”The Aztecs were known to master many technologies, but I don’t have any idea what this could be”, Dr. Jones commented. You inspected the object and noticed that there was a grid carved into the surface. At that moment, several cards with skulls fell close to your foot. Curiously enough, each card would fit into each cell of the grid. ”That’s weird, it looks like a very macabre game, but I guess it has to do with the riddle,” you said. ”I am sure it does!”, Dr. Jones replied.

You heard the deep voice again. ”If you solve this riddle, I will open the door for you. I will give you two numbers, let us call them c and r. You need to place the cards over the grid cells such that the number of cards in every column is equal to c and the number of cards in every row is equal to r. Once you find a way, you say out loud One”.

”Okay, this does not seem to be a very hard riddle”, you thought. ”The vaults problem looked much harder”. You started to imagine how this card assignment would look like. Figure 1 shows an example for a grid of size 2x3 for which the total number of cards in each row and each column is 2.

But the deep voice continued. ”Once you are done, I will check your card assignment. Your goal is to find another distinct card assignment in the grid, such that the row and column requirements are fullfilled. If you find it, say out loud Two. Repeat the process, stating how many distinct card assignments you have found.” ”This is not going to be easy”, you commented. By doing some simple math (you did quite well in Discrete Maths, by the way), you noticed that the number of distict card assignments can be very large for the grid object in your hands. ”Once you find the total number, stop and call me. I will check if the total number is correct. If it is, I will open the door for you. Otherwise, you will stay in this chamber for the rest of your life.”

To make their lives easier, your goal is to develop a program that, given a grid of a certain size, reports the total number of distinct card assignments according to the column and row requirements described above.

Input: The input starts with one line containing one integer T that corresponds to the number of test cases. Then, two lines follow. The first line contains the number of columns (nc) and rows (nr) of the grid. The next line contains the values for c and r.

Output: For each test case, you should print the total number of distinct card assignments in a grid of size nc x nr such that the total number of cards in every row is r and the total number of cards in every column is c. This number is always less than 2^{63}.

Nome: ____________________________________ Nº de estudante: ______________

C) **Problema C** (1.5 pontos)

Considere uma variante do problema C em que existem G *floodgates* disponíveis, e 0 *manhole covers*. De forma semelhante ao problema original, o objetivo é encontrar posições para os G *floodgates* que permitam bloquear a água e assim criar um caminho seguro. Considere que existe sempre uma solução válida. Para além disso, considere que o número de *floodgates* G é mínimo, ou seja não existe nenhuma solução válida que utilize menos *floodgates* que G.

Descreva uma abordagem que permita resolver esta variante, e indique a complexidade temporal da mesma. Serão valorizadas abordagens que tenham menor complexidade temporal.

Problem C

Description

Once you found the total number of distinct card assignments, you hear the deep voice once again, “Well done, you may pass”. You go through the door and find yourself in a new room. In the center of this room you see a model of a maze made up of square spaces. On one of those spaces, at the end of the maze, there is what seems to be the exit! You are almost out!

You also notice that there is a door in front of you, and you see the same door at the bottom of the model maze. Dr Henry Jones notices this too and says: “Hmm, it seems we will enter the maze on the model if we go through the door. It seems we can use this model to guide us to the exit”. “But look sir”, you say, “it seems that in some locations there are some manholes. I suspect that once we go through the door, the maze will start filling with water from these manholes!”

Dr Jones, looking at an inscription below the model says “It seems you are right, this inscription here says ‘Block the water to be free’”. “And what is this?”, you ask. You are holding what seems to be several model manhole covers, and one model flood gate. When you place them on the model, or take them out, you hear noises coming from the maze, as if what you are doing in the model is also happening inside the maze.

“If my translation is correct”, says Dr Jones looking at a manuscript he found near the model, “we can use the manhole covers to stop water from coming out of the manholes, and the flood gate to stop water from crossing between two spaces. I think that we need to place these objects in the model to create a safe path from the start of the maze to the exit”. You agree, and start working on a program to help with this task.

Goal: Given a maze, find where to place the flood gate and manhole covers to create a safe path that can go from the start of the maze to the exit without crossing any water. You must also return this path. It is guaranteed that there is always at least one way to create a safe path. You can consider that water only flows in directions up, left, down or right (never diagonally), and cannot cross walls. Moreover, you assume that every place in the maze reachable by water from an uncovered manhole is immediately filled with water once you enter the maze.

Input: The first line of the input gives the number of test cases. There are at most 10 test cases per input. The first line of each test cases gives two space-separated integers $1 \leq N \leq 500$ and $1 \leq M \leq 500$ denoting the number of rows and columns used to represent the maze. The following N lines give the maze. Each line contains M characters. Each character is one of:

\# A wall
. An empty space
E The exit (this is always on the first line)
D The door where you will start (this is always on the last line)
M A manhole where water will come from once you open the door

Lastly, there is one line with an integer C denoting the number of manhole covers you have available. It is guaranteed that C is less than the number of manholes in the maze (which means that you must always use the flood gate).

Output: For each test case you must print several lines. The first line must contain four space-separated integers r_1,c_1,r_2,c_2 which will give the location of the flood gate. In particular, this means that the flood gate will stop water from crossing from (r_1, c_1) to (r_2, c_2) and vice-versa. The two spaces must be adjacent. The second line starts with an integer $0 \leq X \leq C$ denoting the number of manhole covers used. The following X lines two space-separated integers denoting the coordinates r_i, c_i of the i-th manhole cover. A manhole cover must be placed on a manhole. The next line starts with an integer $P \geq 2$ denoting the number of steps to get from the door to the exit. The following P lines give the path. In particular, each line contains two integers denoting the coordinates r_j, c_j of this path in the order they are visited. The first coordinates should be the coordinates of the door, and the last should be the coordinates of the exit. The safe path cannot go over walls or cross the flood gate. There can be no repeated coordinates in this path, that is, you cannot go over the same place twice in the safe path. Consider that the top-left corner of the map has coordinates (0, 0).