

Exegese - Mosaico de Fotos de Satelite

Geffté Luis de Souza Caetano

¹ Faculdade de Computação

Universidade Federal de Mato Grosso do Sul (UFMS) – Campo Grande – Brasil

{geffte.caetano}@ufms.br

1. Introdução

Quando foi proposto desenvolver um programa que envolvesse receber dados que representassem pontos no plano cartesiano que formassem um retângulo e comparar esses retângulos uns com os outros para verificar o menor possível nos quais os outros caberiam parecia a princípio uma tarefa simples, o que de fato era quando olhando em terceira pessoa o programa pronto, mas quando se senta para fazer de fato e tenta apenas “codar”, desenvolver essa ideia que a princípio era simples, torna-se algo não tão intuitivo. Assim, como o problema tinha certo grau de dificuldade, tive que refletir sobre o problema e pensar em formas de como resolver. Para facilitar, utilizei instrumentos como geogebra e até mesmo o caderno para rabiscar algumas ideias. Isso ajudou a pensar em maneiras de resolver o problema. De fato, desafiador.

2. Sobre a ideia

Analisando o que foi fornecido, é relevante dizer que o crescimento dessa área se dá tanto aos números negativos quanto positivos. Assim é correto afirmar, a fim de analisar a largura do retângulo que como xse é o começo e xid é o fim, para que haja sobreposição um outro retângulo tem que estar dentro desse intervalo ou ser maior que ele. Porém, pensando que o crescimento dos números se estende ao negativo, no caso da coordenada xse , uma outra deve ser menor ou igual a ela. Como a coordenada xid se estende aos números positivos, então uma outra coordenada deve ser maior ou igual.

O mesmo acontece com a altura do retângulo, considerando que yid é o começo e yse é o fim da altura, para que haja sobreposição esse outro retângulo o intervalo deve ser maior. Assim, uma outra coordenada yid deve ser menor ou igual para que se estenda aos números negativos e para o yse deve maior ou igual para que se estenda aos positivos.

Assim foi compreendido o que se fazer para resolver o problema em termos abstratos. Então colocando em termos metódicos, e levando em relevância que esses virão a ser dados em um arquivo, uma estratégia de como fazer de fato foi criada.

3. Sobre a estratégia e implementação

O problema detalha que os dados das coordenadas foram dados em um arquivo .txt, então para que fosse feita a leitura foi usado o comando `with`, que serviu para abrir, ler e fechar o arquivo. Com isso surgiu o primeiro desafio, estava evidente que deveriam ser criadas as variáveis xse , yse , xid , yid , mas como elas seriam instanciadas? Pois como comumente era feito, se fosse atribuído o valor 0, o programa entenderia que aquela coordenada se encontra na origem, e não que não tem valor nulo. Porém, outro problema também estava evidente, essas variáveis seriam para leitura e comparação, mas o problema pede um valor específico da coordenada do retângulo final, e como essas primeiras

seriam usadas para leitura, outras que foram o valor final foram criadas, chamadas xseF, yseF, xidF e yidF. Mas elas também não poderiam ser instanciadas com o valor 0 se não ocorreria o mesmo problema. Assim, considerando que variáveis xseF...yidF seriam utilizadas como valor final, foram instanciadas com a primeira linha que significa o primeiro retângulo. Com fim de comparação, as variáveis xse...yid também foram instanciadas com a primeira linha, porém dentro de um for, para que todos os retângulos fossem comparados.

Dessa forma, o problema de leitura dos valores estava resolvido, então, surge a comparação das variáveis em sim, onde deve ser considerado tanto os crescimento dos números negativos e positivos. Assim, a priori foi if utilizado para fazer essa comparação, porém após algumas pesquisas a função max() e min() faziam bem esse trabalho, e para deixar o código limpo e eficiente, foi feita essa troca. Então, toda vez que uma coordenada xse...yid tivesse o intervalo maior xseF...yidF, era efetuada a atualização de valor das variáveis xseF...yidF. E por fim, mas não menos importante, a impressão das coordenadas é feita.

4. Pontos a serem considerados

Acredito que mesmo após entender o desafio proposto, um problema a ser enfrentado é como começar a desenvolver uma solução. De fato, eu dediquei algum tempo para desenhar no papel o que eu precisaria fazer para comparar os intervalos, mas quando vamos levar isso para o computador sempre surgem alguns imprevistos, como por exemplo, como instanciar as variáveis, afinal se fizesse como de costume e atribuísse o valor 0, eu estaria atribuindo a coordenada de origem para esta variável, então fazer ela receber apenas o valor da primeira linha resolveu tanto o problema das variáveis finais quanto das variáveis de comparação.

Então, eu não diria problema a ser resolvido, mas ponto a ser melhorado, era a comparação entre as coordenadas feita pelo if. Era realmente uma maneira de fazer essa comparação, mas talvez não fosse a melhor, então tentei buscar uma maneira melhor. E realmente pode não ser a melhor que exista, mas é a minha melhor maneira atualmente.

Porém, o maior desafio não foi codar e usar os artifícios da linguagem para resolver o problema, acredito que o maior desafio foi entender o que estava acontecendo e desenvolver uma ideia e posteriormente uma estratégia para o resolver de fato. Como última consideração, desafios como este instigam o pensamento e o desafio clama o progresso, então mesmo que houve algumas dificuldades a serem resolvidas, também houve progresso ao fim desse desafio.