

Relatório 2º Projeto ASA 2023/2024

Grupo: AL20

Alunos: Mariana Santana (106992) e João Rodrigues (106221)

Descrição do Problema e da Solução:

O programa desenvolvido no âmbito deste projeto calcula e devolve o número máximo de saltos entre SCCs do grafo especificado no input. Para isso recorre a uma adaptação do algoritmo de Kosaraju que, ao identificar um novo SCC, calcula o maior número de saltos entre os SCCs já descobertos, a partir do SCC atual.

Na função *main* é efetuada a recolha dos dados n (nº vértices) e m (nº arestas) e a inserção das m correspondências entre vértices em listas de adjacências para o grafo original e transposto. A função principal (*calcMaxSpread*) utiliza a função *DFS1* que modifica a stack *descEndTimes* e a devolve com os vértices ordenados por ordem decrescente de finalização da DFS. Em seguida, é usada a função *DFS2* que modifica o vetor *results* e o devolve com o número máximo de saltos entre SCCs que podem ser executados a partir de cada vértice. Finalmente, é calculado o valor máximo do vetor *results* que coincide com o resultado final.

Análise Teórica:

Pseudocódigo de alto nível da função *calcMaxSpread*:

```
descEndTimes := new stack  
visited := new vector( $n + 1, 0$ )  
scc := new vector( $n + 1, 0$ )  
results := new vector( $n + 1, 0$ )  
  
for  $i$  in  $n$   
  if non visited[ $i$ ]  
    DFS1( $i$ ) → add  $i$  to descEndTimes  
  
reset visited to 0  
  
while descEndTime is not empty  
   $v$  := descEndTimes.top()  
  if non visited[ $v$ ]  
    DFS2( $v$ ) → set SCCs and calculate its results  
  
 $\text{max} := \text{max}(\text{results})$ 
```

Recolha e processamento dos dados de entrada, globalmente $O(m)$:

Leitura dos valores de n e m e das correspondências de m arestas (ciclo que depende linearmente de m): $O(m)$;

Inserção de cada correspondência entre vértices nas listas de adjacências do grafo: $O(1)$;

Aplicação do algoritmo para cálculo e preenchimento do vetor *results*, globalmente $O(n + m)$:

DFS1 – No pior caso todos os vértices e todas as arestas são visitadas e consultar valores em vetores e as operações com as stacks são $O(1)$: $O(n + m)$;

DFS2 – Analogamente à DFS1, sendo que os cálculos e propagações dos resultados dentro da SCC dependem do nº de vértices desse SCC e não contribuem para a complexidade total: $O(n + m)$;

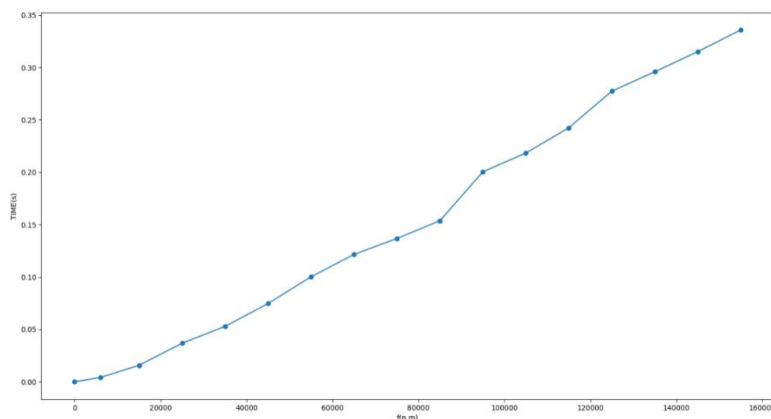
Cálculo resultado máximo: $O(n)$;

Apresentação do resultado: $O(1)$:

Complexidade global da solução: $O(m) + O(n + m) + O(1) = O(n + m)$.

Avaliação Experimental dos Resultados:

Para a elaboração deste gráfico realizamos 17 experiências com diferentes valores de n e m e nº de SCCs do grafo, como exposto na tabela e tivemos em conta a variação do tempo em função da quantidade $f(n, m) = n + m$ e verificamos que essa variação é linear, como mostrado nas figuras abaixo.



Nº SCCs	n	m	f(n,m)	TIME(s)
0	0	0	0	0
200	1000	5000	6000	0.004165
400	5000	10000	15000	0.0158
800	10000	15000	25000	0.036858
1200	15000	20000	35000	0.052889
1400	20000	25000	45000	0.07471
2000	25000	30000	55000	0.10019
3000	30000	35000	65000	0.121565
5000	35000	40000	75000	0.136796
7000	40000	45000	85000	0.153742
10000	45000	50000	95000	0.200414
12000	50000	55000	105000	0.218294
15000	55000	60000	115000	0.242477
18000	60000	65000	125000	0.277361
20000	65000	70000	135000	0.295868
23000	70000	75000	145000	0.315101
25000	75000	80000	155000	0.33574