

### Implementação 3

Allan Guilherme Gomes Pego

Implementei um método que constrói grafos direcionados simples aleatórios(sem arestas antiparalelas e de ciclos unitários) de tamanhos distintos (número de vértices 100, 1.000, 10.000, 50.000), cria-se um grafo mais esparsos na primeira iteração (arquivos a1, b1, c1, d1) e um mais denso na segunda(arquivos a2, b2, c2, d2).

Foram utilizados como referência externa os códigos de Sedgewick: FlowEdge, FlowNetwork, FordFulkerson, Queue e Bag. Estes foram adaptados para receber um forwardStar com um grafo para encontrar o fluxo máximo. Após este passo, itera-se pela lista de arestas do sumidouro até encontrar uma aresta que tenha fluxo, após este passo se faz o caminho contrário até a fonte.

No momento da criação do grafo utiliza-se a estrutura de lista de adjacência, no momento que estes arquivos são salvos em disco em um arquivo binário, a estrutura utilizada é o forwardstar.

São encontrados os caminhos disjuntos do mesmo arquivo por dez vezes, estes testes são salvos em um arquivo CSV informando a iteração, tempo em milissegundos, os números de fonte e sumidouro, número de vértices, número de arestas, número de caminhos disjuntos.

Existe também o código em Python para geração dos gráficos com média, mediana e média ponderada.

O main está na classe EncontraCaminhosDisjuntos,

Os arquivos utilizados no teste estão anexados a esta entrega, caso queira gerar novos é só tirar o comentário do método geraOsGrafosAleatorios().

