

Escreva um programa que leia um arquivo passado como parâmetro de entrada, e gere um grafo a partir deste arquivo. Seu programa deve permitir:

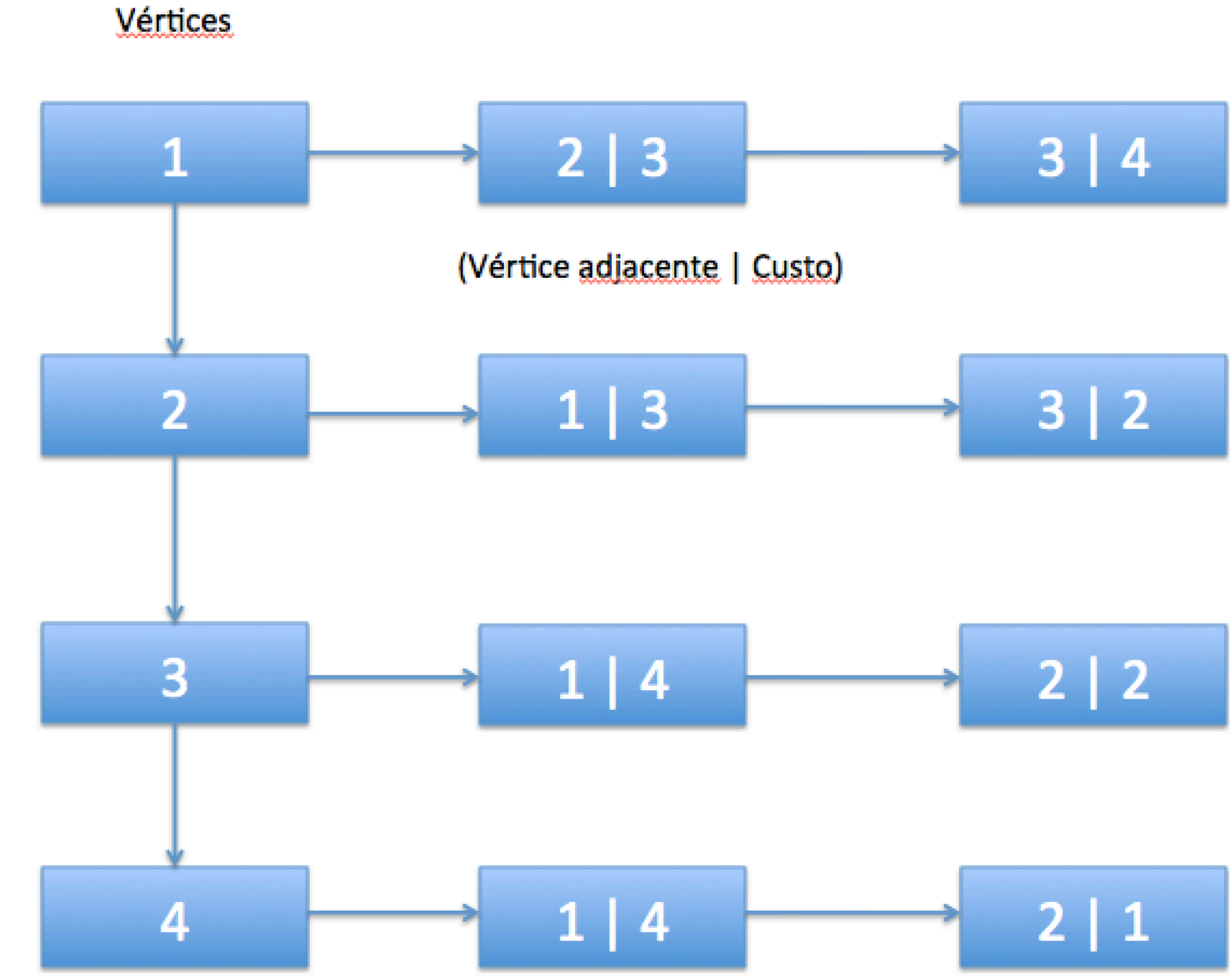
1. Inserção e retirada de nós e arestas;
2. Verificar se o grafo é conectado. Se a resposta for verdadeira, seu programa deve realizar:
 1. Calcular e gravar no arquivo agm.txt a árvore geradora mínima (usando Prim ou Kruskal); e
 2. Calcular e gravar no arquivo cmc.txt o caminho mais curto de um nó (passado como parâmetro de entrada) a todos os demais, usando Dijkstra
3. E, finalmente, seu programa deve gravar no arquivo novo.txt o grafo resultante das modificações realizadas pelo usuário.

Exemplo de entrada.txt:

```
4 (número de vértices)
1 (nome do vértice)
2
3
4
6 (número de arestas)
1 2 3 (aresta do vértice 1 para o vértice 2, com custo 3)
1 3 4
1 4 1
2 3 2
2 4 1
3 4 2
```

Representação em forma de lista:

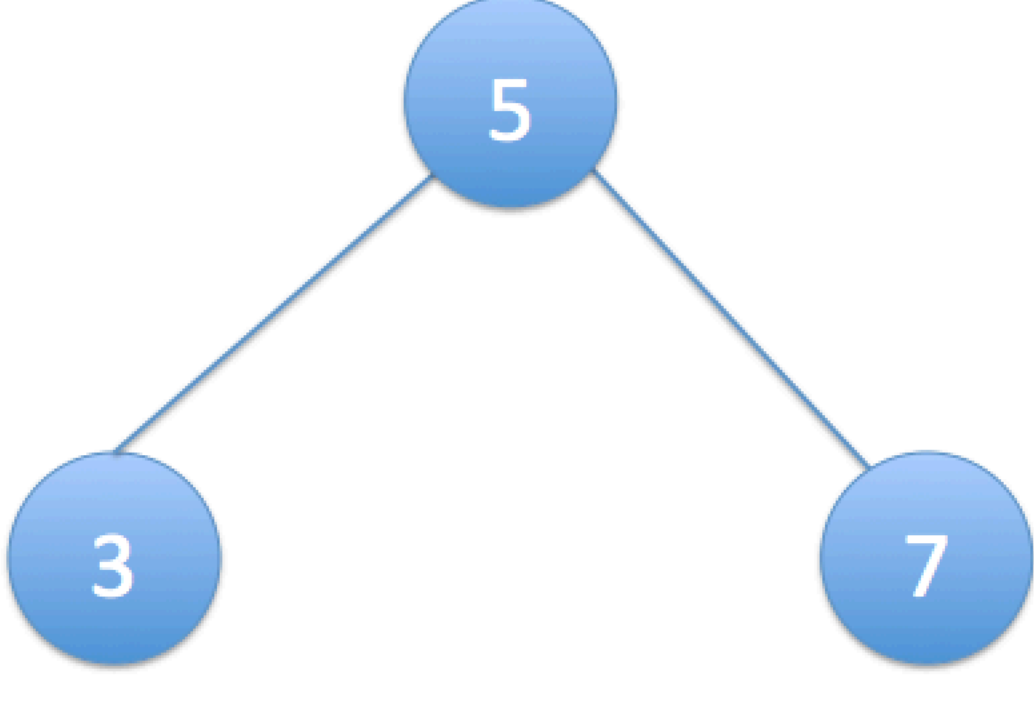
Dois ponteiros, um para os vértices e outro para as adjacências desse vértice.



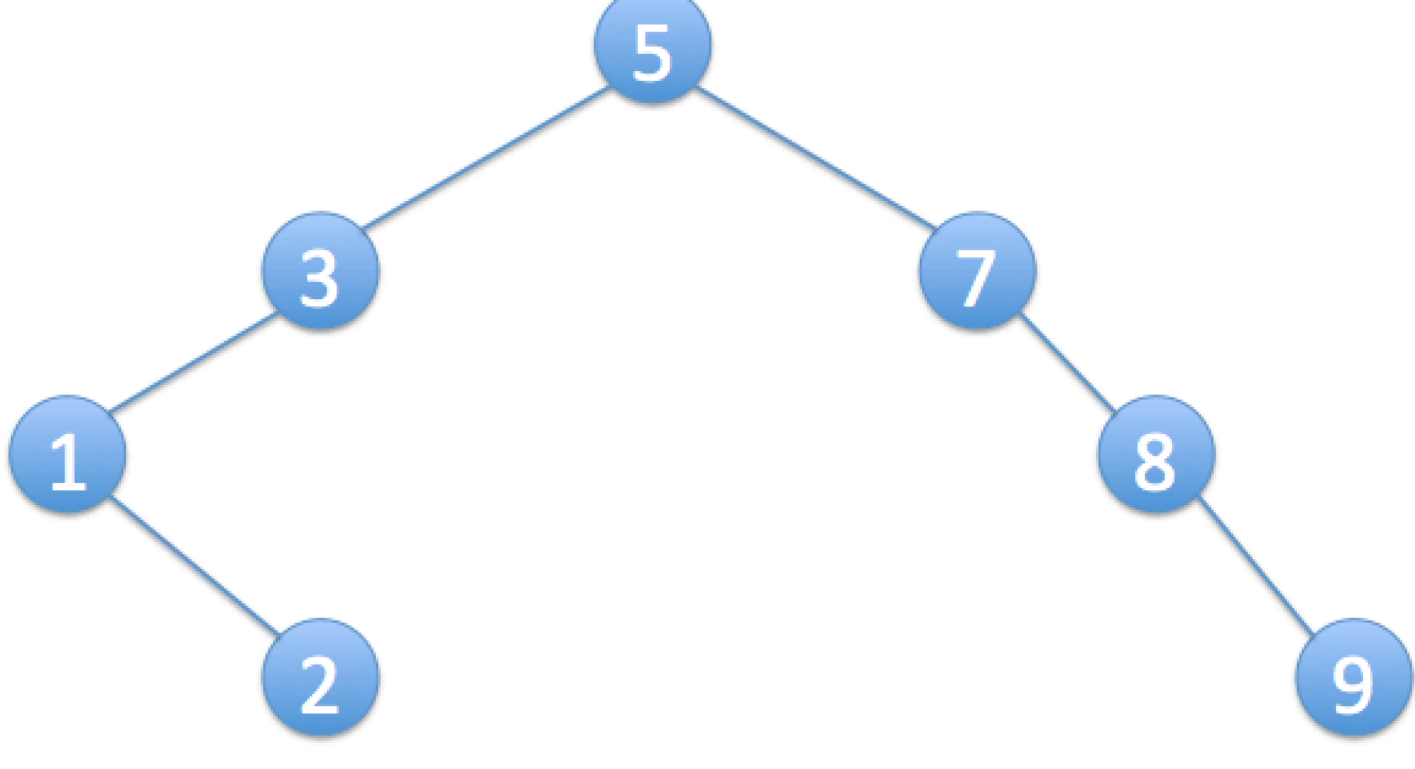
(não coloquei todas do exemplo acima por falta de espaço =D)

Árvore AVL

- Em ABB existe uma relação de ordem
- Operações de balanceamento: inserção e retirada
- Ex:



Inserir 1,2,8,9:



- Solução: AVL e AVP

1. AVL

- Todas as operações são $O(\log(n))$: Busca, inserção e retirada
- Fator de balanceamento (FB), Altura (Esquerda), Altura (Direita)
 - $FB(a) = \{-1, 0, 1\}$
- Altura (número máximo de passos para chegar em uma folha) (Código)

2. Rotações

- Existem 4 rotações:
 - RSE: $FB(P) = -2$; $FB(R) = -1$

