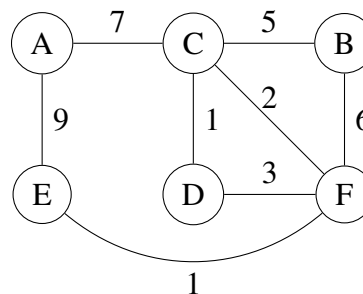
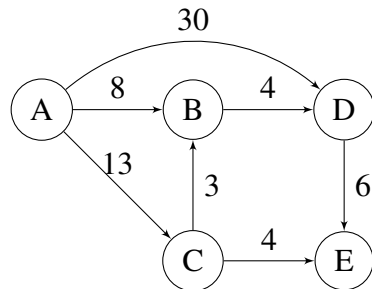


UFPE/CIn – ENGENHARIA DA COMPUTAÇÃO
IF672 – AED 2018.2 – EXERCÍCIO ESCOLAR 2
PROFESSOR: GUSTAVO CARVALHO
NOME: _____

1. {2,0 pt.} Seja G um grafo dirigido acíclico com n nós, representado como uma matriz de adjacências (0 indica a ausência de aresta e 1 a presença de aresta), escreva um código para: `void toposort(int[][] G, int n)`. Este código deve imprimir no final uma ordenação topológica do grafo G .
2. {2,0 pt.} Considerando o algoritmo de Dijkstra (usando uma *heap*), e o grafo abaixo, calcule os menores caminhos a partir do nó A. Mostre a evolução do array de distâncias (inicialmente, com ∞ para todos os nós exceto A) e da heap como um array (inicialmente, só com o par (A,0)) após a visita de cada nó do grafo.
3. {2,0 pt.} Considerando o algoritmo de Kruskal (usando uma *union-find*: *quick-union* sem compressão), calcule a árvore geradora de peso mínimo para o grafo abaixo. Mostre também a evolução da *union-find* como um array (inicialmente, um conjunto para cada nó). Ao fazer uma união, a raiz deve ser o nó representativo lexicograficamente menor.



4. {2,0 pt.} Seja $A = \{3, 5, 6, 7, 8\}$, encontre *todos* os $A' \subseteq A$, tal que a soma de seus elementos seja igual a 15. Resolva o problema usando *backtracking* (filho à esquerda/direita = inclusão/exclusão do i -ésimo item, respectivamente). Desenhe a árvore de espaço de estados. Os nós devem ser rotulados com o somatório atual. Ao não continuar a busca em certo nó, explique o porquê.
5. {2,0 pt.} Considerando uma mochila com capacidade de 4 kg, e os itens (peso, valor): $i_1 = (2, 20)$, $i_2 = (1, 10)$, $i_3 = (3, 20)$, $i_4 = (2, 15)$, encontre o subconjunto de itens mais valioso que cabe na mochila. Use programação dinâmica (*bottom-up*) e apresente a matriz (*item* \times *capacidade*) construída na busca.