

Estruturas de Dados II

G5

Grafos

1 – Implemente um grafo usando listas ligadas de modo que cada nó de cabeçalho encabece duas listas: uma contendo os arcos emanando do nó de grafo e outra contendo os arcos terminando no nó de grafo.

2 – Implemente um grafo de modo que as listas de nós de cabeçalho e nós de arco sejam circulares.

3 – Implemente um grafo usando um vetor de listas de adjacência. Sob essa representação, um grafo de n nós consiste em n nós de cabeçalho, cada um contendo um inteiro de 0 a $n - 1$, e um ponteiro. O ponteiro é para uma lista de nós de lista, cada uma das quais contendo o número de um nó adjacente ao nó representado pelo nó de cabeçalho. Implemente o algoritmo de Dijkstra usando essa representação em grafo com o vetor organizado num heap ascendente.

4 - Considere o grafo do quadro abaixo. O programa *schedule* dá saída na seguinte organização de tarefas:

Tempo	Subtarefas
1	A, B, C
2	D, E
3.....	F
4.....	G

Essa organização exige três auxiliares (para o período de tempo 1). Você pode descobrir um método de organizar as subtarefas de modo que somente dois auxiliares sejam necessários em qualquer período de tempo e mesmo assim o serviço inteiro possa ser feito nos mesmos quatro períodos de tempo? Escreva um programa que organize as subtarefas de modo que um número mínimo de auxiliares seja necessário para finalizar o serviço inteiro no número mínimo de períodos de tempo.

5 – Se existir somente um trabalhador disponível, levará k períodos de tempo para finalizar o serviço inteiro, onde k é o número de subtarefas. Escreva um programa para listar uma seqüência válida na qual o trabalhador possa executar as tarefas. Observe que esse programa é mais simples do que o *schedule* porque não será necessária uma lista de saída; assim que o campo *count* chegar a 0, a tarefa poderá ser eliminada. O processo de converter um conjunto de precedências numa única lista linear, na qual nenhum elemento posterior preceda um anterior, é chamado **ordenação topológica**.