

Lista de Exercícios

1. O que é um *grafo de dependência de tarefas* e qual sua importância no desempenho de um sistema paralelo?
2. Para cada um dos grafos de tarefas apresentados na figura abaixo, calcule:
 - (a) O grau máximo de concorrência.
 - (b) O tamanho de caminho crítico.
 - (c) O máximo *speedup* possível se houver um número arbitrário de processadores disponíveis.
 - (d) O número mínimo de processos necessário para atingir esse *speedup* máximo.
 - (e) O *speedup* máximo caso estejam disponíveis apenas (i) 2, (ii) 4 ou (iii) 8 processadores.

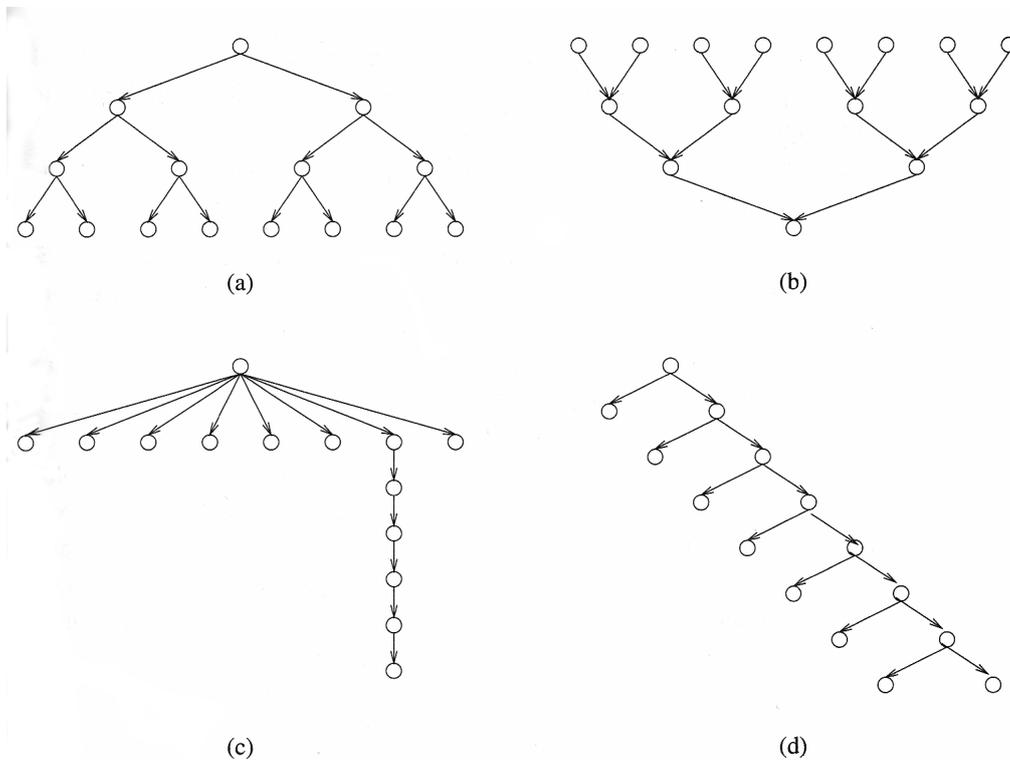


Figura 1: Grafos de dependência de tarefas para a questão 1.

3. O que é *granulosidade (granularity)* e qual a sua relação com o *grau de concorrência*?
4. O que é um grafo de interação de tarefas?
5. Um programa paralelo de multiplicação de matrizes vai calcular $\mathbf{C} = \mathbf{A} \cdot \mathbf{B}$ com matrizes de 2000×2000 elementos e roda com 16 tarefas organizadas em uma grade bidimensional 4×4 , com tarefas identificadas por sua posição (i, j) na grade. Cada tarefa (i, j) irá calcular uma submatriz de elementos contíguos de \mathbf{C} das linhas $500i$ a $500(i + 1) - 1$ e colunas de $500j$ a $500(j + 1) - 1$ (linhas e colunas numeradas de 0 a 1999). Inicialmente, cada tarefa possui os valores das matrizes \mathbf{A} e \mathbf{B} com os mesmos índices dos valores da matriz que será calculada (isto é, as três matrizes são distribuídas identicamente).
Monte um diagrama do grafo de interação de tarefas deste algoritmo.

6. Considere o programa de, dado um vetor (grande) de números inteiros e um valor inteiro, encontrar todos os índices do vetor que tenham o valor especificado. Proponha algoritmos paralelos para esse problema usando:
 - (a) Decomposição de dados.
 - (b) Decomposição recursiva.
 - (c) Decomposição híbrida (recursiva + de dados).
7. Dados dois conjuntos de valores $A = \{a_0, a_1, \dots, a_n\}$ e $B = \{b_0, b_1, \dots, b_m\}$ e um predicado $p(a, b)$ (isto é, uma função com valores verdadeiro ou falso), queremos calcular uma lista de todos os pares de valores com um elemento da lista A e um da lista B para os quais o predicado é verdadeiro, isto é, queremos $\{(a, b) \mid a \in A, b \in B, p(a, b)\}$. Descreva quatro formas de paralelização desse problema e faça uma comparação entre elas, considerando que n e m são grandes, mas no caso geral $n \ll m$.
8. Comente sobre o impacto de comunicações entre tarefas com estruturas irregulares no mapeamento de tarefas (em contraposição a quando as estruturas são regulares).
9. Em que situações uma distribuição em blocos é mais adequada do que uma distribuição cíclica? Em que situações a distribuição cíclica é mais adequada que a em blocos? Como a distribuição bloco-cíclica se posiciona em comparação com as outras duas mencionadas?
10. Compare vantagens e desvantagens de esquemas centralizados versus distribuídos para mapeamento dinâmico de tarefas.
11. Defina os seguintes termos relacionados com o desempenho de sistemas paralelos:
 - (a) Tempo de execução paralelo.
 - (b) Tempo total paralelo.
 - (c) *Speedup*.
 - (d) Sobrecarga (*overhead*) paralela.
 - (e) Eficiência.
 - (f) Custo paralelo.
12. O que é um sistema paralelo de custo ótimo?
13. Como se pode quantificar a escalabilidade de um sistema paralelo usando a função de isoeffiência?
14. Como se define o *speedup* escalado com limitação de tempo? E com limitação de memória? Por que essas considerações são importantes?
15. Considere um problema para o qual o tempo de execução paralela de um certo sistema paralelo é¹

$$T_p = \Theta \left(t_c \frac{n^2}{p} + t_s \log p + t_w \frac{n}{\sqrt{p}} \log p \right),$$

onde n é o tamanho do problema, p o número de processadores e t_s e t_w constantes do sistema de comunicação. Este problema faz uso de $\Theta(n^2)$ memória e um algoritmo sequencial para sua solução tem tempo de execução $\Theta(t_c n^2)$. Encontre:

- (a) O *speedup*.
- (b) A eficiência.
- (c) O custo paralelo.
- (d) A sobrecarga paralela.
- (e) A função de isoeffiência.
- (f) O *speedup* escalado com limitação de tempo.
- (g) O *speedup* escalado com limitação de memória.

¹Na análise assintótica de algoritmos paralelos, como os tempos das constantes de computação e de comunicação podem ser muito diferentes, é costume manter os fatores de crescimento mais rápido multiplicando cada uma dessas constantes.

16. Para a solução paralela de um mesmo problema, dispomos de três algoritmos paralelos, cujos tempos de execução são:

$$\begin{aligned}T_1 &= t_c \frac{n^3}{p} + t_s \sqrt{p} + t_w \frac{n^2}{\sqrt{p}} \\T_2 &= t_c \frac{n^3}{p} + t_s \log p + t_w \frac{n^2}{\sqrt{p}} \\T_3 &= t_c \frac{n^3}{p} + t_s \log p + t_w \frac{n^2}{p^{2/3}} \log p\end{aligned}$$

O problema usa $\Theta(n^2)$ memória e um algoritmo sequencial leva tempo $\Theta(n^3)$. Faça uma análise comparativa desses três algoritmos usando a função de isoefficiência e os *speedups* limitados por tempo e memória.