

Estruturas de Dados — MC202ABC
1º Semestre 2018
Instituto de Computação — UNICAMP
4ª Lista de Exercícios

Entregue os exercícios até 01/07/2018 pelo SuSy.
Leia com cuidado as instruções abaixo.

Instruções:

- Nesta lista, escolha **apenas dois** dos três exercícios propostos para resolver. Exercícios resolvidos a mais não contarão como nota extra.
- Os exercícios deverão ser entregues separados em tarefas diferentes no SuSy: o exercício 1 deve ser entregue na tarefa **14ex1**, o exercício 2 deve ser entregue na tarefa **14ex2** e assim por diante. Caso não faça algum dos exercícios, basta não submeter na tarefa referente ao mesmo.
- Se preferir, apresente algoritmos como um pseudocódigo ao invés de uma implementação em C, isto é, descreva o seu algoritmo com uma lista de passos muito bem definidos, inclusive usando estruturas como **if** e **while**.
- A nota da lista será dada em relação aos três exercícios resolvidos, apesar da submissão ocorrer em três tarefas diferentes do SuSy.
- Os exercícios podem ser digitados em um computador (preferencialmente) ou serem escritos à mão e digitalizados (você pode inclusive tirar uma foto). Porém, a entrega precisa ser feita necessariamente por meio de um arquivo **PDF** com tamanho de no máximo **2MB**. Outros formatos *não serão aceitos*.
- Garanta que o documento tenha qualidade o suficiente para ser lido. Documentos escritos à mão devem ter letras legíveis e feitos preferencialmente à caneta, já que o lápis tem pouco contraste para a digitalização. Assim, se possível, opte por fazer o exercício no computador ao invés de digitalizar.
- Caso tenha problemas com o tamanho do arquivo gerado, procure por um compressor de PDFs online. Existem vários serviços que recebem um PDF e geram o mesmo PDF com um tamanho menor. Porém, tome cuidado de garantir que o arquivo continue legível.
- Lembre-se de indentar corretamente o seu código para facilitar o entendimento.
- **Não se esqueça de colocar nome e RA em cada exercício.**
- A correção de cada exercício será enviada para o seu email institucional (da DAC). Garanta que há espaço na sua cota para receber os emails¹ e verifique o spam se necessário.

¹Considere fazer o redirecionamento do seu email da DAC para o seu email pessoal: <https://www.dac.unicamp.br/portal/estudantes/webmail-mais-informacoes>.

Exercício 1 (14ex1): Escreva uma função que, dado uma lista L de chaves inteiras não-negativas, encontra inteiros positivos a e M , com M mínimo, tal que a função de hashing $h(x) = a \cdot x \pmod M$ é injetora para a lista de chaves dada, isto é, para quaisquer duas chaves diferentes $y \in L$ e $z \in L$, $h(y) \neq h(z)$. Dessa forma, $h(x)$ é um hashing perfeito. **Não** é necessário desenvolver o algoritmo em C, basta apresentar um pseudocódigo com a ideia.

Exercício 2 (14ex2): Chamamos de circuito um grafo conexo tal que todo vértice nesse grafo está conectado a exatamente duas arestas. Mais formalmente, o grafo C é um circuito se é conexo e $d(v) = 2$, $\forall v \in V(C)$. Denotamos por C_k um circuito com k vértices, para $k \geq 3$. Observe os exemplos de circuitos na imagem 1.

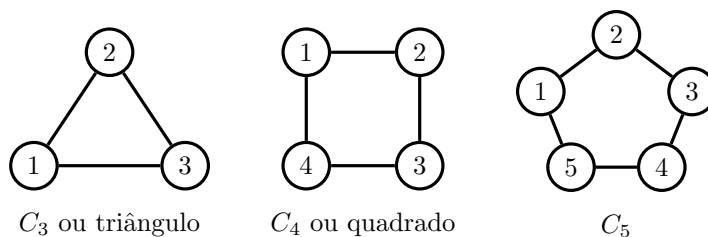


Figura 1: Exemplos de circuitos

Dado um grafo simples $G = (V, E)$, faça um algoritmo com complexidade de tempo $O(|V|^3)$ para decidir se G contém um C_4 , ou seja, decidir se existe um subgrafo $H \subseteq G$ que é um C_4 . **Não** é necessário desenvolver o algoritmo em C, basta apresentar um pseudocódigo com a ideia.

A Figura 2 apresenta um exemplo de grafo que possui um C_4 .

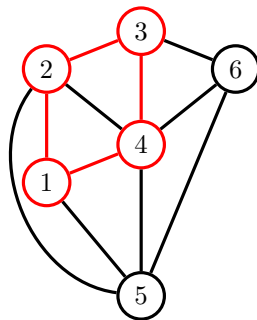


Figura 2: Exemplo de grafo que contém C_4 (formado pelos vértices e arestas em vermelho).

Exercício 3 (14ex3): A distância entre dois vértices u e v de um grafo G , denotada por $dist(u, v)$ é dada pelo custo do menor caminho de u até v em G . Caso não exista um caminho ligando u a v , $dist(u, v) = \infty$.

Dado um grafo $G = (V, E)$ em que todas as arestas possuem peso 0 ou 1, faça um algoritmo com complexidade de tempo $O(|V| + |E|)$ para calcular a distância entre um dado vértice $S \in V(G)$ e os demais vértices de G . **Não** é necessário desenvolver o algoritmo em C, basta apresentar um pseudocódigo com a ideia.