

Primeira Prova de Análise e otimização de Código
- DCC888 -
Ciência da Computação

Nome: _____
“Eu dou minha palavra de honra que não trapacearei neste exame.”

Número de matrícula: _____

As regras do jogo:

- A prova é sem consulta.
- Quando terminar, não entregue nada além do caderno de provas para o instrutor.
- Quando escrever código, a sintaxe correta é importante.
- Cada estudante tem direito a fazer uma pergunta ao instrutor durante a prova. Traga o caderno de provas quando vier à mesa do instrutor.
- A prova termina uma hora e quarenta minutos após seu início.
- Seja honesto e lembre-se: **você deu sua palavra de honra.**

Alguns conselhos:

- Escreva sempre algo nas questões, a fim de ganhar algum crédito parcial.
- Você pode sacrificar sua pergunta para saber como morreram 37 ladrões na estória de Ali-Babá e os Quarenta Ladrões.
- Se não entender a questão, e já tiver gasto sua pergunta, escreva a sua interpretação da questão junto à resposta.
- A prova não é difícil, ela é divertida, então aproveite!

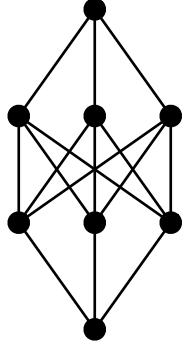
Tabela 1: Pontos acumulados (para uso do instrutor)

Questão 1	Questão 2	Questão 3	Questão 4

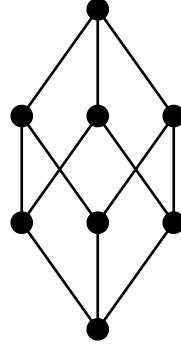
Questão extra - 0.5 Pontos - escreva o nome de um livro de um autor que já ganhou o prêmio Nobel de literatura.

1. Esta questão refere-se ao conceito de reticulado.

a)



b)



(a) (5 Pontos) Considere os dois diagramas acima. Um desses diagramas poderia representar um reticulado. O outro não. Qual desses diagramas representa um reticulado? Por que o outro não representa reticulado algum?

(b) (5 Pontos) Seja L um conjunto parcialmente ordenado por uma operação \leq . É sabido que:

- Se todo subconjunto de $Y \subseteq L$ possui um menor limite superior $\sqcup Y$
então
- todo subconjunto de $Y \subseteq L$ também possui um maior limite inferior $\sqcap Y$

Prove que essa afirmação é verdadeira. Dica:

$$\sqcap Y = \sqcup \{l \in L \mid \forall l' \in Y : l \leq l'\}$$

2. (10 Pontos) Um código é dito isócrono com relação a uma entrada s quando seu tempo de execução é sempre o mesmo, para qualquer possível valor de s . Nesta questão assumiremos que um programa é isócrono quando o resultado de seus desvios condicionais não é influenciado por s ¹. A figura abaixo mostra dois programas, um isócrono, e outro não isócrono:

Programa não isócrono

```
int findKey1(int* s, int k) {
    int i;
    for (i = 0; i < 80; i++) {
        if (s[i] == k)
            return 1;
    }
    return 0;
}
```

Programa isócrono

```
int findKey2(int* s, int k) {
    int i;
    int hasFound = 0;
    for (i = 0; i < 80; i++) {
        hasFound |= (s[i] == k);
    }
    return hasFound;
}
```

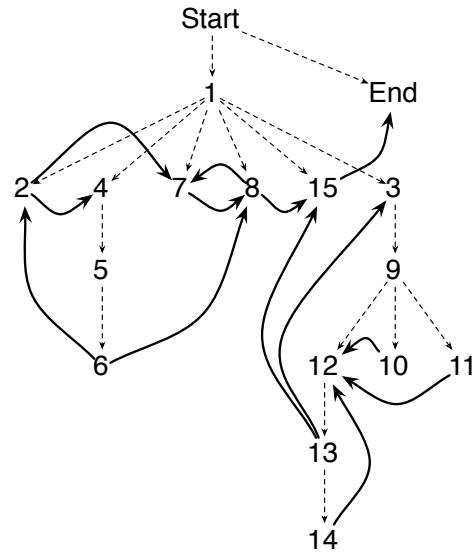
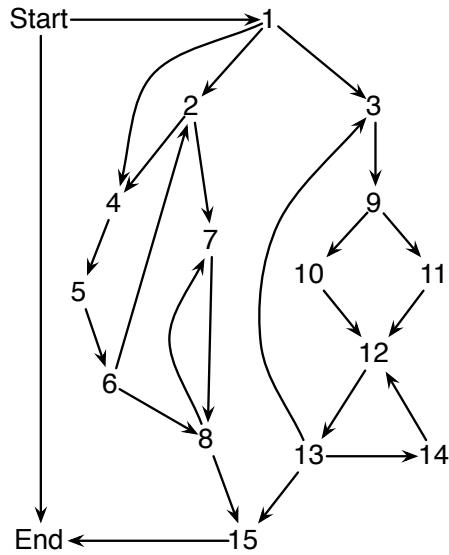
É possível que otimizações de código transformem um programa sem desvios condicionais isócronos em programa não isócrono. Nesta questão pede-se:

- Um exemplo de programa P sem desvios condicionais.
- Um novo exemplo, que poderia ser produzido por um compilador como `gcc`, que transforme o programa P da questão acima em um programa com pelo menos um desvio condicional não isócrono.
- Uma breve descrição da otimização usada. Você precisa usar uma otimização conhecida. Em outras palavras, você precisa usar uma otimização que possa existir em `gcc` ou `LLVM` com chance razoável.

As respostas para os três items acima serão avaliadas em conjunto – não há avaliação individual de cada item dessa questão.

¹Na verdade, existem outras situações, como falhas de acesso a cache, que também são fontes de não isocronia.

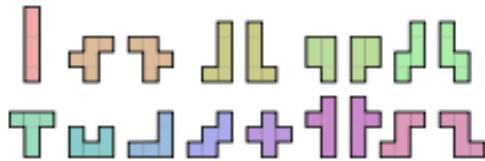
3. Esta questão refere-se às duas figuras abaixo. Essas figuras foram retiradas do artigo: “A Linear Time Algorithm for Placing for placing ϕ -Nodes”, publicado em POPL’95 por Sreedhar and Gao. Assuma que a figura da esquerda represente o grafo de fluxo de controle de um programa, e cada nodo represente um bloco básico:



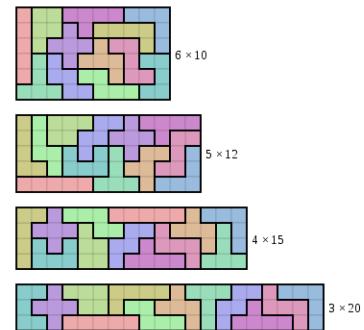
- (a) (2 Pontos) Qual o significado das arestas pontilhadas, no grafo da direita?
- (b) (3 Pontos) Qual o significado das arestas sólidas no grafo da direita?
- (c) (3 Pontos) Imagine que o nodo 14, no CFG da esquerda, contenha uma definição de uma certa variável v . Em que outros nodos a conversão dessa definição para o formato de Atribuição Estática Única vai exigir a criação de funções ϕ no programa da esquerda?
- (d) (2 Pontos) Assuma, agora, que além da definição de v no nodo 14, tenhamos somente um uso dessa variável no nodo 12 do CFG da esquerda. Neste caso, a técnica de poda (*pruning*) pode levar à criação de menos funções ϕ . Em quais nodos deveriam ser inseridas funções ϕ , dadas as suposições acima, e dado que estamos criando o formato *pruned SSA*?

4. Pentaminós são figuras geométricas formadas pela união de cinco quadrados, de forma que quadrados adjacentes compartilhem uma aresta inteiramente. A figura abaixo, à esquerda, mostra os possíveis pentaminós:

Os 12 possíveis pentaminós. Peças assimétricas são mostradas espelhadas



4 diferentes quebra-cabeças construídos com pentaminós



Um quebra-cabeça de pentaminós consiste em arranjar todos os 12 ladrilhos em uma forma pré-definida. Acima, à direita, vemos quatro diferentes quebra-cabeças. Nesta questão, você deve assumir a existência de um algoritmo A , que resolva um quebra-cabeças de pentaminó. O algoritmo A lê duas entradas: um tabuleiro parcialmente preenchido, e um conjunto de peças ainda por serem arranjadas. Este conjunto é chamado de *banco de peças soltas*. Ele retorna o tabuleiro totalmente preenchido com as peças que estavam originalmente soltas. A não realiza *back-tracking*. Em outras palavras, dada a configuração do jogo, A sempre faz um movimento correto, isto é, que leva a uma solução, se houver uma solução possível. Um movimento consiste em arranjar exatamente uma peça sobre o tabuleiro, possivelmente retirando uma ou mais peças que já estavam lá para o banco de peças soltas.

(a) (5 Pontos) Enumere algumas suposições que, se válidas, garantem a terminação do algoritmo A .

(b) (5 Pontos) Enumere algumas suposições que, se válidas, garantem que A encontra corretamente uma solução para o quebra-cabeças, assumindo que uma solução exista.