

Resolução – Lista 10 (Projeto e Análise de Algoritmos)

Março - 2023 / Leticia Bossatto Marchezi – 791003

Questão 1

Sabe-se que o problema da parada (the halting problem) é incomputável. Explique esse problema discutindo porque ele é incomputável.

Resolução:

The Halting Problem é um problema em que temos um programa que deve parar caso encontre uma certa entrada, caso contrário deve continuar rodando infinitamente.

Considerando uma função g tal que:

```
def g():  
    if halts(g):  
        loop_forever
```

Entretanto, não há algoritmo que resolva o problema, pois caso a função $\text{halts}()$, que decide se a subrotina deve parar, retorne True, então o programa entrará em loop infinito, e caso retornar False o programa será finalizado pois a condição do if não foi satisfeita. Ou seja, em ambos os casos chegam em contradições, e não é possível resolver este problema.

Questão 2

Sobre classificação de problemas, responda:

Resolução:

(a) Explique o que são problemas polinomiais, ou seja, pertencentes a classe P. Cite um exemplo de problema P

Problemas P-complexos são problemas que podem ser resolvidos com complexidade polinomial, ou seja, $O(n^k)$, e são razoavelmente rápidos. Estes problemas podem ser resolvidos por máquinas de turing determinísticas.

Um exemplo de algoritmo da classe P são os algoritmos de busca em árvores.

(b) Explique o que são problemas não-determinísticos polinomiais, ou seja, pertencentes a classe NP. Cite um exemplo de problema NP.

Problemas Não-determinísticos polinomiais são problemas em que é fácil provar que uma solução é válida, mas difícil de construir uma solução otimizada para ele. São necessárias máquinas de turing não determinísticas para resolvê-los.

Um exemplo de problema NP é o problema da mochila, ou Knapsack problem.

Questão 3

Ainda sobre classificação de problemas, responda:

Resolução:

(a) Explique o que são problemas não-determinísticos polinomiais completos, ou seja, pertencentes a classe NP-Completo. Cite um exemplo de problema NP-Completo.

Problemas NP-completos são baseados em duas definições: - Se o problema é NP e pode ser reduzido ao tempo de execução $O(n^k)$, ou seja, é redutível a um problema de ordem polinomial, então é NP-completo - Problemas NP-Completo também são problemas NP-Hard que estão inclusos em NP.

Um exemplo de problema NP-Completo é o Problema do caixeiro viajante.

(b) Explique o que são problemas pertencentes a classe NP-Hard. O que os difere de problemas NP-completos? Cite um exemplo de problema NP-Hard.

Problemas NP-Hard são problemas de otimização, e não decisão, que são no mínimo tão difíceis quanto qualquer problema NP-completo. Além disso, há problemas NP-Hard em que não é viável testar se uma certa solução é válida.

Um exemplo de problema NP-Hard é o Problema do caixeiro viajante.

Questão 4

Explique em detalhes o que é a redução de um problema.

Resolução:

A redução de um problema é um método de estender uma solução de um algoritmo para adaptá-lo e resolver outro problema.

Tendo um algoritmo A_1 que soluciona um problema L_1 , e outro problema L_2 que se busca resolver transformando o algoritmo A_1 para uma solução A_2 . Dado um input x , A_1 retornará True se x é um input válido para L_1 e False caso contrário. Para resolver a redução é tomado um valor de entrada y para L_2 que é aplicado em uma função f transformadora, convertendo-o em um input para A_1 . Assim, A_1 retornará True ou False para a função $f(y)$, que corresponderá à solução de L_2 para y .

Questão 5

Pesquise sobre o Teorema de Cook-Levin (é permitido perguntar ao Google e ao Chat-GPT) e descreva-o brevemente. Porque ele é importante? Qual a implicação desse resultado teórico?

Resolução:

O teorema de Cook-Levin provou que o Problema de satisfatibilidade booleana (SAT) faz parte dos problemas NP-completos. Ou seja, pertence ao conjunto dos problemas NP. Dessa forma, qualquer problema NP pode ser solucionado em tempo polinomial ao ser reduzido ao SAT por uma máquina de Turing determinística.

A implicação do resultado é que, se há uma solução em tempo polinomial para o SAT, então também deve existir uma solução em tempo polinomial para qualquer problema NP que pode ser verificado em tempo polinomial.