

LISTA OBRIGATÓRIA PARCIAL DE PROJETO E ANÁLISE DE ALGORITMO

SEMANAS 13 e 14

Para as questões a seguir, faça:

- Descreva sua ideia de solução;
 - Elabore o algoritmo em pseudo-linguagem;
 - Implemente a solução.
- Um k -dígito é um número composto com k dígitos, $1 \leq k \leq n$. Por exemplo, 193 seria um 3-dígito e 14783 seria um 5-dígito. Se o primeiro dígito for 0, ele não é considerado válido. Assim 01276 não é um 5-dígito válido. Esta restrição ocorre apenas para o primeiro dígito, logo 10010 é um 5-dígito válido. Elabore um algoritmo para gerar todos os n -dígitos válidos tal que a soma dos dígitos das posições pares seja igual à soma dos dígitos das posições ímpares. Para $n=3$, por exemplo, os dígitos a seguir seriam determinados.

110 121 132 143 154 165 176 187 198 220 231 242 253 264 275 286 297 330 341 352
363 374 385 396 440 451 462 473 484 495 550 561 572 583 594 660 671 682 693 770
781 792 880 891 990
 - Considere o conjunto $S=\{s_1, s_2, \dots, s_n\}$ de n inteiros positivos. Usando a técnica de *backtracking*, determine o conjunto potência de S . O conjunto potência é formado por todos os subconjuntos de S , incluindo o próprio S e o conjunto vazio. Por exemplo, para o conjunto $\{1, 5, 9\}$, o conjunto potência é dado por $\{\{\}, \{1\}, \{5\}, \{9\}, \{1,5\}, \{1,9\}, \{5,9\}, \{1,5,9\}\}$. Na sua solução, além do que se pede acima nos itens (a) a (c), construa a árvore de estados do problema como item (d).
 - Elabore um algoritmo para resolver problema da Mochila usando a técnica de *branch-and-bound*. Dica: consulte na seção 12.2. do Levitin uma possível formulação para o problema. Na sua solução, além do que se pede acima nos itens (a) a (c), construa a árvore de estados do problema como item (d).
 - Suponha que você tenha uma função que gere números aleatórios no intervalo de 1 a n , chamada *random(1,n)*. Considere que você tenha um vetor de inteiros, ordenado em ordem crescente, com mesma quantidade de números pares e ímpares. Elabore um algoritmo probabilístico para embaralhar os números desse vetor de tal forma que no vetor resultante não haja três números consecutivos de mesma paridade. Qual método você está usando? Justifique.
 - Suponha que seja dado a você uma função *escolha()*, que retorna 0 com 60% de probabilidade e 1 com 40% de probabilidade. Usando apenas essa função, elabore outra função *escolhaJusta()* que retorna 0 com 50% de probabilidade e 1 também com 50% de probabilidade.