

Prova P2
Algoritmos e Estrutura de Dados II
2021/2 - 05/05/2022

Nome: ~~XXXXXXXXXX~~
Matrícula: ~~XXXXXXXXXX~~

Questão 1 (2,5 pontos)

Modifique o mínimo possível o algoritmo de backtracking mostrado abaixo para geração de subconjuntos de um conjunto V de números inteiros, tal que a soma dos elementos do subconjunto impresso é ímpar.

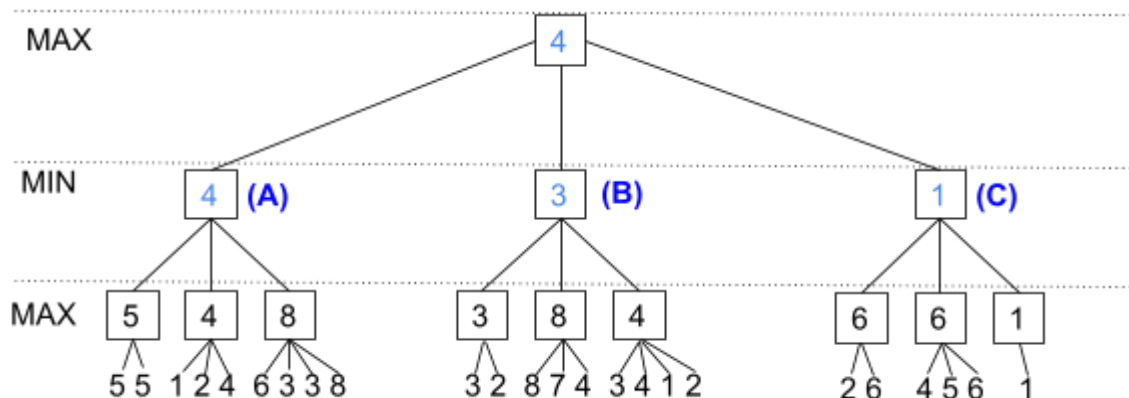
```
GeraSub (ns, t):  
para i ← t..n incl.:  
    S[ns] ← V[i]  
    escreve(S)  
    se (i<n)  
        GeraSub (ns+2, i + 1)
```

```
para i ← 1..n incl.:  
    ler V[i]  
GeraSub (1, 1)
```

Questão 2 (2 pontos)

Considere a árvore de jogo ilustrada a seguir. Note que é a vez de MAX jogar e cada folha apresenta a pontuação estimada por um avaliador estatístico heurístico.

- Preencha os nós em branco com os valores MAXMIN adequados.
- Informe qual é a melhor jogada para MAX, escolhendo entre A, B e C.
Entre A, B e C, a melhor jogada é A (4).
- Qual a pontuação que MAX espera alcançar?
MAX espera alcançar 4.



Questão 3 (3 pontos)

Suponha que você fará uma viagem de carro entre Belém/PA e o Rio de Janeiro, passando, principalmente, pelas rodovias federais. Para se preparar para a longa viagem, você garantiu que tem em seu navegador GPS o mapa do trajeto contendo a distância entre todos os postos de combustível em sua rota. Suponha que n seja um valor dado e corresponda a distância em quilômetros percorrida por seu carro quando ele está com o tanque cheio.

Escreva um algoritmo guloso para determinar em quais postos de combustível você deve parar, de modo a fazer o menor número de paradas ao longo do trajeto. Argumente porque o seu algoritmo é guloso e que a solução produzida é ótima.

Algoritmo:

n #distância em km que o carro anda com o tanque cheio

$n+1$ #posição em que o carro fica SEM combustível

postoCombustivel(n):

posicaoAtual = posicaoInicial

enquanto (posicaoAtual < posicaoFinal):

 se ($n+1$ < posicaoFinal):

 encontrar o posto mais próximo antes de chegar na $n+1$

 abastecer nesse posto mais próximo

 posicaoAtual = posto

 senao:

 Posição atingida!

Fim do algoritmo

Provando por contradição:

Assumindo que o algoritmo não encontre uma solução ótima, ou seja, existe uma solução melhor. A minha solução é $C1 = \{pc1, pc2, pc3, \dots, pck\}$ e a melhor solução é $C2 = \{pc1', pc2', pc3', \dots, pck-1'\}$.

Comparando as paradas nos postos, começando pela primeira, há 3 possibilidades:

$pci = pci'$: Continua a comparar as próximas paradas.

$pci > pci'$: Então $C1$ é maior. Sem afetar o resultado, podemos fazer $pci' = pci$ e guardar este para o caso anterior.

$pci < pci'$: De acordo com o algoritmo, isso acontece toda vez que selecionamos por último o possível posto antes de ficar sem combustível. pci' então é maior que pci .

Se $pci = pci'$ para todas as paradas $k-1$, $C1$ faz mais uma parada pck para evitar ficar sem combustível. $C2$ não faz essa parada -> o carro ficará sem combustível -> $C2$ é inválido.

Assim, provamos que, para todos os casos, $C2$ é inválido -> uma solução melhor não existe. Então, $C1$ é uma solução ótima -> o algoritmo acima, portanto, é ótimo.

Questão 4 (2,5 pontos)

Construa uma árvore de Huffman para as seguintes símbolos com suas respectivas frequências, onde NLPN = número de letras do seu primeiro nome e NLUS = número de letras do seu último sobrenome:

NLPN = M E R C I A = 6

NLUS = M A R Q U E S = 7

(F, 13)

(A, 20)

(N, NLUS=7)

(B, 9)

(M, NLPN=6)

Ordenando:

(A, 20)

(F, 13)

(B, 9)

(N, NLUS = 7)

(M, NLPN = 6)

O passo a passo para construir a árvore de Huffman será:

