

1. (10 pontos) O que é e como funciona um *Algoritmo Guloso*? Explique com exemplos.
2. Seja $P(n, S)$ um problema de otimização com n variáveis $x_1, x_2, \dots, x_n$ tais que $x_i \in S$, $|S| \geq n$, e com a restrição de que cada valor em S seja atribuído a uma única variável, ou seja, $x_i \neq x_j$ para todo $i \neq j$. O objetivo é maximizar a função $f : \{(x_1, x_2, \dots, x_n) \mid x_i \in S\} \rightarrow \mathbb{R}$, que é desconhecida.
- (a) (15 pontos) Proponha um algoritmo de Backtracking para este problema. Explique como deve ser feita a ramificação, escolha de valores e cortes por viabilidade.
- (b) (15 pontos) Suponha que existe um algoritmo que resolve eficientemente o problema de encontrar o máximo da função objetivo f do problema $P(n, S)$ com $x_i \in S$ sem garantir que a restrição $x_i \neq x_j$ seja atendida. Este algoritmo pode ter alguma utilidade para resolver $P(n, S)$? Justifique.
3. Considere a seguinte recorrência que descreve as soluções de um problema R em função de subproblemas:

$$R(i, j) = \begin{cases} 0, & \text{se } i \geq j, \\ \max\{R(i, m-1) + R(m+1, j) + m \mid i \leq m \leq j\}, & \text{caso contrário.} \end{cases}$$

Onde queremos resolver o problema $R(1, n)$, com $n \geq 1$, e v é uma função de $\{1, \dots, n\}$ para os naturais. Responda:

- (a) (10 pontos) Os subproblemas se repetem?
- (b) (10 pontos) É possível usar a técnica de *Backtracking* para resolver R ? É preciso alguma adaptação? Qual seria? Como seria a ramificação e a escolha de candidatos?
- (c) (15 pontos) É possível usar a técnica de *Branch & Bound* para resolver R ? É preciso alguma adaptação? Qual seria? Como poderia ser uma função limitante?
- (d) (15 pontos) É possível usar a técnica de *Programação Dinâmica* para resolver R ? É preciso alguma adaptação? Qual seria? Qual seria a estrutura de dados usada para guardar os subproblemas resolvidos e qual seria a ordem de execução?
- (e) (10 pontos) É possível usar a técnica de *Algoritmos Gulosos* para resolver R ? É preciso alguma adaptação? Qual seria?¹
- (f) **Bônus** (10 pontos) Você considera que existe uma função “oráculo” para resolver R usando a técnica de *Algoritmos Gulosos*? Se sim, diga como seria. Se não, esboce uma justificativa para a não existência de tal função.

¹Considere que a existência de tal algoritmo está sempre condicionada a existência ou não de uma função “oráculo”. Você deve responder assumindo que tal função existe.

Gabarito

1. Algoritmos gulosos podem ser descritos de 2 formas (aqui uma delas já basta):

A primeira é focada na resposta. Supondo que a resposta esperada é uma sequência de variáveis, $(x_1, x_2, \dots, x_n)$, o algoritmo atribui o valor de cada uma destas variáveis, em sequência, em alguma ordem e sem voltar atrás. Ou seja, supondo que ordem é a ordem dada pela sequência, seria algo como “para cada i de 1 a n , faça $x_i \leftarrow \text{escolha}(i)$ ”. Onde $\text{escolha}(i)$ é uma função (tipicamente polinomial) que usa os dados da instância e os valores anteriormente atribuídos para $x_1, x_2, \dots, x_{i-1}$ para escolher o valor de x_i .

A segunda é focada em uma recorrência, com escolhas na ramificação, ou seja, que a resposta esperada está em uma das folhas. Assim, o algoritmo escolhe, sem voltar atrás, um dos filhos de cada nó da árvore de recorrência, em sequência, até chegar na folha que é a resposta.

2.

- (a) Em cada nó da árvore de recorrência temos o valor i que indica que as variáveis $x_1, x_2, \dots, x_{i-1}$ já tem valores e que temos que escolher o valor para x_i . Para os cortes de viabilidade, fazemos o conjunto de candidatos C ser $S \setminus \{x_1, x_2, \dots, x_{i-1}\}$ (ou seja, tiramos os valores já usados). Assim, cada nó tem $|C|$ filhos. Isso garante a viabilidade. As folhas são os nós em que $i = n + 1$. E neste ponto calculamos $f(x_1, x_2, \dots, x_n)$ e escolhemos o maior até o momento.

- (b) Podemos usar este algoritmo para construir uma função limitante, já que ele resolve uma versão relaxada do problema original.

Obs.: Faltaria garantir que temos como calcular a função objetivo (da versão relaxada) levando em conta as variáveis já atribuídas.

3.

- (a) Certamente se repetem. Esta recorrência gera repetições pois qualquer valor de m pode ser escolhido, gerando intervalos (i, j) repetidos.
- (b) Sim, sem adaptações. A ramificação é tal que em cada nó descrito por (i, j) , temos um nó filho para cada m , com $i \leq m \leq j$.
- (c) Sim, sem adaptações. Uma função limitante $B(i, j)$ deve ser tal que $B(i, j) \geq R(i, j)$. Uma função limitante simples poderia ser $B(i, j) = \sum_{m=i}^j m$.
- (d) Sim, sem adaptações. A estrutura de dados seria uma matriz $n \times n$, apenas o triângulo superior (com $i \leq j$). A ordem de cálculo é da diagonal para o canto superior esquerdo. Ou da coluna esquerda para a coluna direita, e em cada coluna, da diagonal para cima. Ou da linha de baixo para a linha de cima, e em cada linha, da diagonal para a direita.
- (e) Sim, sem adaptações. Em cada passo decidimos qual m é o escolhido. Note que mesmo assim precisamos de 2 subproblemas para cada problema.
- (f) Uma função oráculo poderia ser escolher $m = j$.

Extra: notem que a recorrência $R(i, j)$ dá SEMPRE a soma $\sum_{m=i+1}^j m = \frac{(i+j+1)(j-i)}{2}$. Sendo assim ela poderia ser substituída por esta expressão.

Na resposta do item (c) temos que $B(i, j) = R(i, j) + i$.

A resposta do item (f) parece ser a única escolha possível, ou seja, deve ser sempre escolher $m = j$.