

1) Considere dois conjuntos de números inteiros A e B com m e n elementos, respectivamente. Eles não estão necessariamente ordenados, e assumamos também que $m \leq n$. Utilizando a estratégia de divisão e conquista, mostre como podemos computar $A \cup B$ e $A \cap B$ em um tempo proporcional a $O(n \lg m)$.

2) Dada uma matriz de números inteiros positivos e negativos, escreva e implemente um algoritmo que encontre um retângulo com a maior soma contido na matriz. A soma de um retângulo é a soma de todos os elementos dentro do retângulo.

Neste problema, o retângulo com a maior soma é referido como maximal. Um retângulo é qualquer submatriz $(n \times m)$, $1 \leq n, m \leq N$, localizado dentro da matriz.

Como exemplo, o retângulo maximal da matriz:

| | | | |
|----|----|----|----|
| 0 | -2 | -7 | 0 |
| 9 | 2 | -6 | 2 |
| -4 | 1 | -4 | 1 |
| -1 | 8 | 0 | -2 |

Será

| | |
|----|---|
| 9 | 2 |
| -4 | 1 |
| -1 | 8 |

Com soma = 15

3) Existe a necessidade de avisar um colega de vocês que determinado professor marcou prova. Para isso, será utilizado um canal extremamente rápido mas que possui um custo de utilização consideravelmente alto, logo, é interessante que sejam utilizados dados com compactação para diminuir o custo do envio. Para fazer a compactação, utilize o Código de Huffman para as frequências:

| | | | | | | | | | | | | | |
|-------------|----|----|---|---|---|---|---|---|---|---|---|---|---|
| caracter: | a | e | h | f | m | n | i | o | v | p | s | r | d |
| frequencia: | 15 | 10 | 4 | 2 | 6 | 3 | 6 | 8 | 2 | 1 | 2 | 1 | 4 |

Assim que construída a árvore de Huffman, codifiquem a seguinte mensagem:

"aprovafoiadiada".

4) Suponha dado um conjunto de livros numerados de 1 a n . Suponha que o livro i tem peso $p[i]$ e que $0 < p[i] < 1$ para cada i . Considere o problema de acondicionar os livros no menor número possível de envelopes de modo que cada envelope tenha no máximo 2 livros e o peso do conteúdo de cada envelope seja no máximo 1.

Escreva um algoritmo guloso $\text{Min-Env}(p, n)$ que recebe um vetor $p[1..n]$ e devolve o número mínimo de envelopes. O consumo de tempo do seu algoritmo deve ser $O(n \lg n)$. Procure mostrar que seu algoritmo está correto.