Algorithms

Assignment: Dynamic Programming

Name: .................................................................

Id: .................................................................

<table>
<thead>
<tr>
<th>Grade</th>
</tr>
</thead>
</table>

Good Luck!
1. Greedy is not optimal for the multiplying $n$ matrices problem.

Define a greedy rule to find the last multiplication for the multiplying $n$ matrices problem.

What is the complexity of the algorithm implied by your greedy rule? Explain.

Construct an example (a sequence of matrices) for which your greedy algorithm fails to find the optimal multiplication order.
2. Let $X = (x_1, \ldots, x_n)$ be a sequence of $n$ real positive numbers.

The set of indices $I \subseteq \{1, \ldots, n\}$ is independent if $|i - j| > 1$ for any two indices $i, j \in I$. That is, there is no index $k \in \{1, \ldots, n - 1\}$ such that both $k$ and $k + 1$ are in $I$.

The weight of a set $J \subseteq \{1, \ldots, n\}$ is the sum of the corresponding $x_i$: $w(J) = \sum_{i \in J} x_i$.

An independent set $I$ is maximum if $w(I)$ is greater or equal to $w(J)$ for any independent set $J$.

The goal is to find such a set for a given sequence $X$.

(a) Find the maximum independent set of the following sequence: $(13, 34, 22, 12, 17, 40, 15, 11, 23)$.

(b) Prove that if $x_i > x_{i-1} + x_{i+1}$ then $i \in I$.

(c) Prove that there is no index $2 \leq k \leq n - 1$ such that $k - 1$, $k$, and $k + 1$ are not in $I$. 


Assume the following greedy solution. Initially let $I$ be empty and $J = \{1, \ldots, n\}$. The set $J$ is the set of indices that are not in $I$ and differ by more than 1 from any index in $I$. That is, $|i - j| > 1$ for any indices $i \in I$ and $j \in J$. Then as long as $J$ is not empty:

(i) **transfer** the index $j \in J$ with the highest value of $x_j$ to $I$.

(ii) **omit** the indices $j - 1$ and $j + 1$ from $J$ if they are in $J$.

(d) Show an example of a positive sequence for a **small** $n$ for which this greedy algorithm fails to produce an optimal solution.

(e) Prove that this greedy algorithm finds a set $I$ whose weight is at least half of the weight of a maximum independent set.
Define the following dynamic programming solution. Let $W(k)$ be the weight of the maximum independent set for the prefix sequence $X_k = (x_1, \ldots, x_k)$ for $1 \leq k \leq n$.

Note that $W(n)$ is the answer to the original problem.

(f) Define the initial values and the recursive formula for $W(k)$.

(g) Describe a bottom-up dynamic procedure to compute $W(\cdot)$.

(h) What is the complexity of your procedure as a function of $n$? Justify your answer.
Let $B(k)$ be TRUE if the index $k$ belongs to the maximum independent set for the prefix sequence $X_k$.

(i) Modify your dynamic programming to compute the function $B(\cdot)$ as well. Describe only the modification.

(j) Describe an efficient procedure to find a maximum independent set for the sequence $X$ based on the function $B(k)$.
3. The coin changing problem.

There are \( n \) coin denominations \( d_n > \cdots > d_2 > d_1 = 1 \). There infinite coins for each denomination. The goal is to pay a positive amount \( A \) with as few as possible coins.

Define a Dynamic Programming optimal solution that finds the number of coins \( n_i \) for each denomination \( d_i \) to get the exact amount: \( A = n_n d_n + n_{n-1} d_{n-1} + n_2 d_2 + n_1 d_1 \). Note that there is always a solution since \( d_1 = 1 \).

(a) Define an objective function, its initial values, and the recursive formula to solve it.

(b) Describe a bottom-up dynamic procedure to compute your objective function.

(c) What is the complexity of your procedure as a function of \( n \) and \( A \)? Should be polynomial in both parameters.